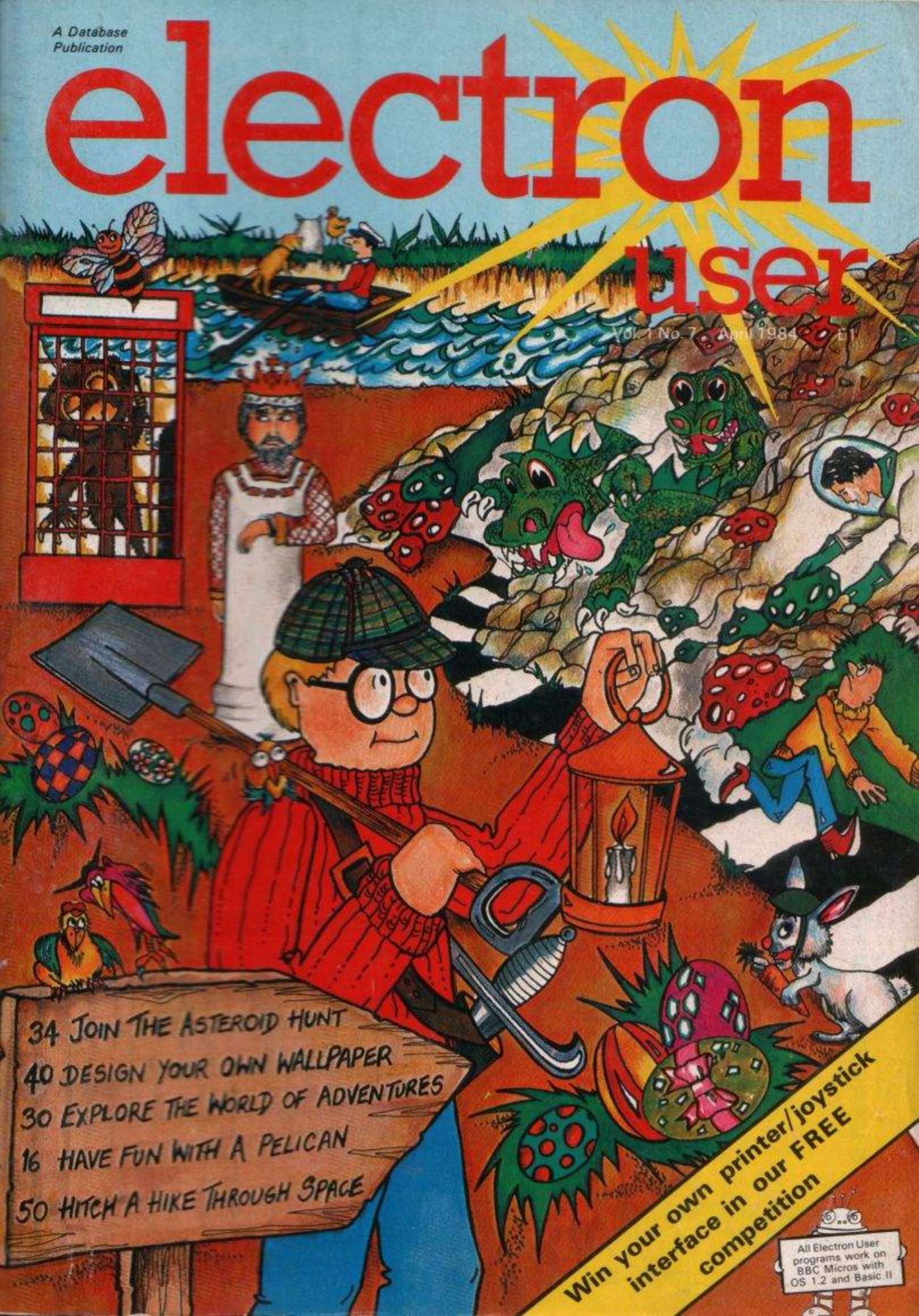
A Database Publication

electron user

Vol. 1 No. 7 April 1984 £1

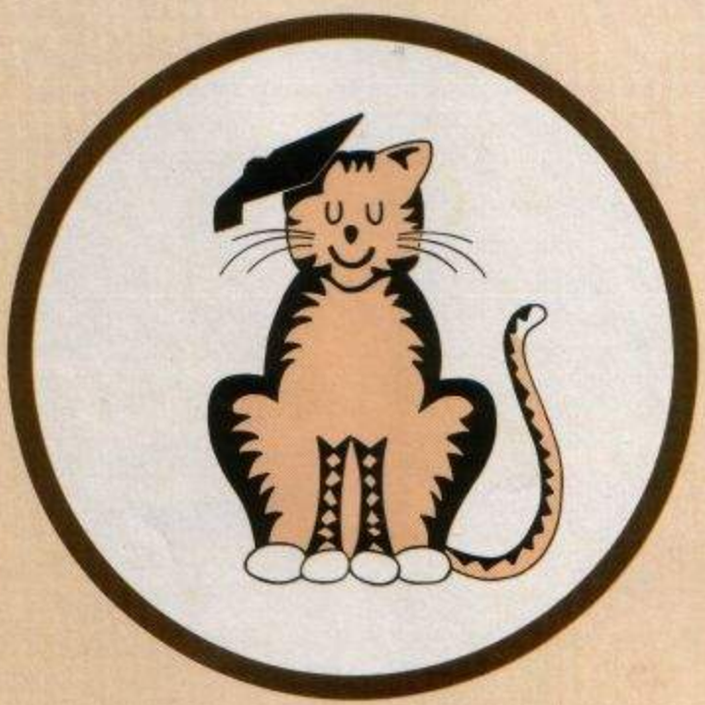


Win your own printer/joystick interface in our FREE competition

All Electron User programs work on BBC Micros with OS 1.2 and Basic II

Exciting News for Electron Users
Now Available

CHESHIRE CAT
EDUCATIONAL SERIES
from
AMPALSOFT



CHESHIRE CAT
The First name in Educational Software.

An exciting range of top quality programs covering all needs from pre-school to 'A' level.

Ampal Computer Services Ltd.
31 Woodbridge Road, Darby Green, Blackwater,
Camberley, Surrey.
Tel: (0252) 876677

Ring 0252 876677
for your nearest stockist

CONTENTS

Vol. 1 No. 7 April 1984

electron user

Managing Editor **Derek Meakin**
Features Editor **Pete Bibby**
Production Editor **Peter Glover**
Layout Design **Heather Sheldrick**
Advertisement Manager **John Riding**
Advertising Sales **John Snowden**
Marketing Manager **Sue Casewell**

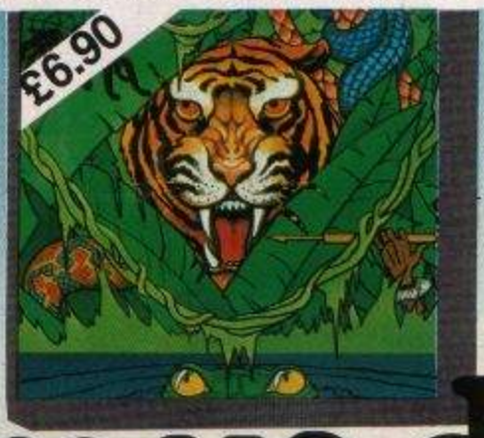
Published by Database Publications Ltd
Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.
Telephone: 061-456 8383 (Editorial) 061-456 8500 (Advertising)
Subscriptions: 061-480 0171 Telex: 667664 SHARET G Prestel: 614568383

Trade distribution in the UK and overseas: Contact Steve Fletcher, Circulation Manager of Database Publications at the above address, or telephone him on 061-480 4153.
Electron User is an independent publication. Acorn Computers Ltd, manufacturers of the Electron, are not responsible for any of the articles in this issue or for any of the opinions expressed.

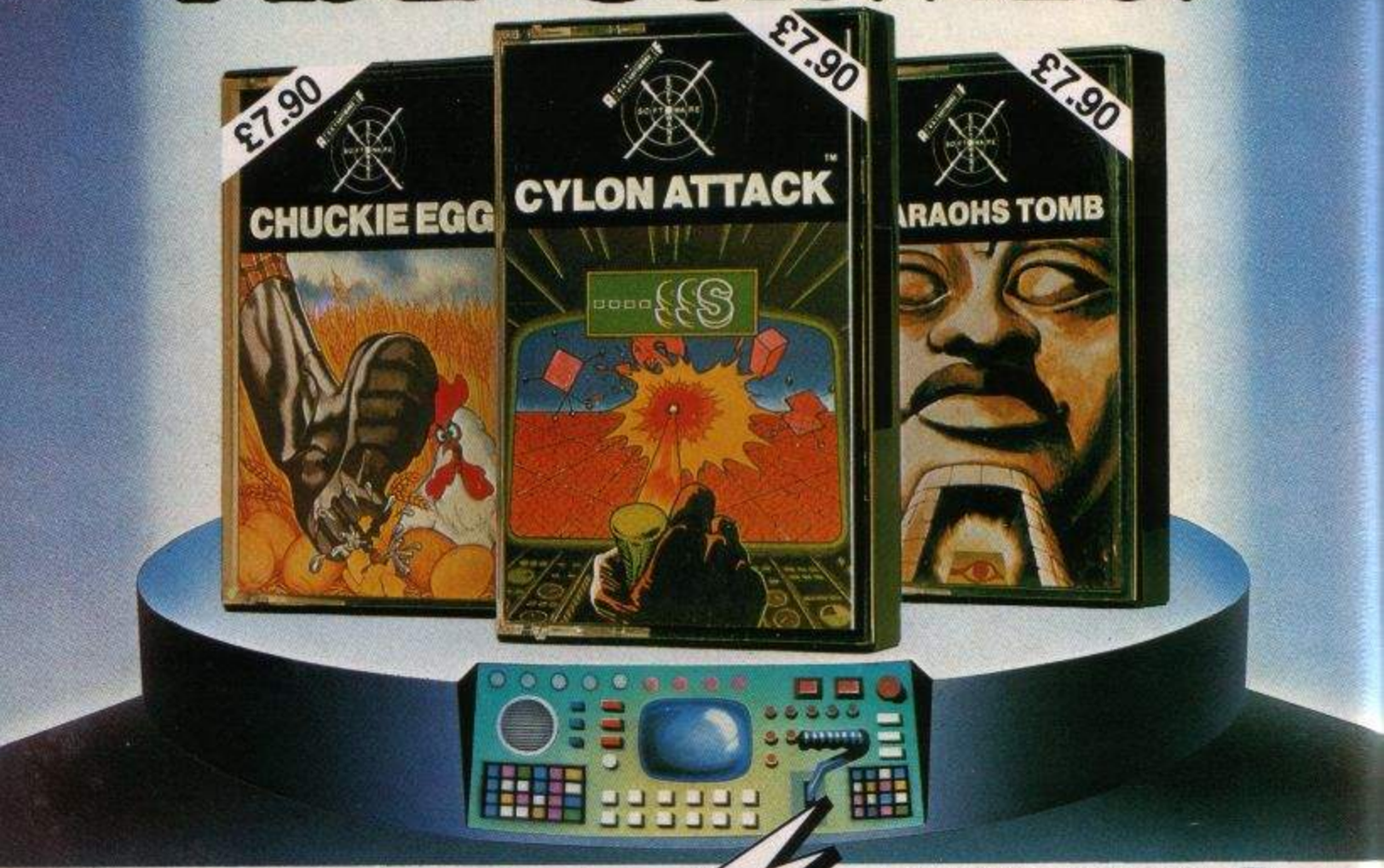
Electron User welcomes program listings and articles for publication. Material should be typed or computer-printed, and preferably double-spaced. Program listings should be accompanied by cassette tape or disc. Please enclose a stamped, self-addressed envelope, otherwise the return of material cannot be guaranteed. Contributions accepted for publication will be on an all-rights basis.

Subscription rates for 12 issues, post free:
£12 UK
£13 Eire (IR £16)
£20 Europe
£20 Rest of world (surface)
£40 Rest of world (airmail)

"ATTENTION EARTHLINGS...



...Beam us down to A&F Software."



	BBC	ELECTRON	DRAGON	SPECTRUM
CHUCKIE EGG	✓	✓	✓	✓
CYLON ATTACK	✓	✓		
JUNGLE FEVER				✓
PHARAOHS TOMB	✓	✓		✓

A&F Software

Available from W.H. Smiths, John Menzies and all leading computer stores.

Unit 8, Canalside Industrial Estate,
Woodbine Street East, Rochdale, Lancs.
OL16 5LB. Tel: 0706 341111

electron user NEWS

Production problems still dog Acorn

Tom Hohenburg

HOPES that Acorn had finally cracked its Electron production problems with the signing up of two additional manufacturers have not materialised.

Rather than more becoming available, in the last few weeks supplies have virtually dried up.

And dealers who believed Acorn's pre-Christmas promises of lots more Electrons going on sale in January have had to tell potential customers that they have no idea when they will be able to meet their orders.

Acute

The problem is getting more acute every day, with orders for the seemingly non-existent machines soaring dramatically.

The total backlog of orders now stands at almost a quarter of a million machines.

Last October Acorn announced that because the Malaysian factory could not produce anything near the number of Electrons needed, a new production line was being set up in Wales.

The firm claimed it would be turning out 4,000 a week from January. They now admit no Welsh-built Electrons will be available until April at the earliest.

They also announced they were setting up a third production line in Hong Kong.

But when *Electron User* spoke to the manufacturers they said that they too would be unable to start shipping them to Britain for another few weeks.

While confirming that the three plants would soon be in full production, Acorn's marketing manager Tom Hohenburg sounded a note of caution:

"With the best will in the world we cannot simply produce hundreds of thousands of machines just like that", he said.

Although Acorn will not give any details, it is understood that one tiny component, a custom-made control device, has been responsible for freezing production on the Electron.

This problem is now said to have been resolved and the production lines are able to move into top gear.

On show at the B-I-G show

THE spring Electron and BBC Micro User Show will see the launch of First Byte Computers' new switched joystick interface for the Electron.

The unit, which allows Electron owners to use any Atari style joysticks, consists of a plug-in cartridge that fits on the expansion board at the back of the micro.

This is only one of many new products that will make their debut at the show, being held at the Royal Horticultural Hall, Westminster, from Thursday March 29 to Sunday April 1.

First Byte has taken

Turn to Page 6

High failure rate

THE problem caused by the shortage of Electrons is being compounded by the unusually high failure rate of machines that have been sold so far.

Dealers contacted by *Electron User* say they have had to return between eight and 25 per cent of the machines they have sold because of faults.

But full marks to Acorn in a difficult situation. They have made it a priority to replace defective machines immediately.

Just think of a game..

DID you know that it might one day be possible to control your Electron by the power of thought alone?

Apparently researchers in behavioural engineering in California are working on games that users can play simply by thinking about what they want to do.

The idea is that the player holds an object that is sensitive to the galvanic skin response, just like lie detectors. Thoughts can affect the conductivity of the skin and variations in this can be used to control the game.

While it may seem to be a lot of trouble to go to in order to play Space Invaders, the research could be of great benefit to the physically handicapped.

Education market booming

THE Electron is following in the footsteps of its big brother, the BBC Micro, by its wide use in education.

More and more schools are now ordering Electrons as additional machines to their BBC Micros.

Software companies are also looking to the new market to increase their sales.

One of the first on the scene are Bourne Educational Software of Hampshire.

They have released three programs aiming to help children develop counting, number recognition and compass skills.

Rewritten especially for the Electron, each of the programs comes with an explanatory booklet.

Although Bourne are an independent company, they are being distributed by Acornsoft following the Acorn subsidiary's new policy of buying in software from other companies.

Not to be left out, Squirrel Software of Manchester have developed a program aimed at helping remedial readers.

Called Visual Recall the software has already proved its worth in extensive testing in schools, helping children with many different kinds of reading difficulties.

From Golem of Bracknell comes Jigsaw Puzzles, a set of six programs for the Electron.

Suitable for children from five to 12 years of age, they were written to help in the development of special concepts and in the formation of problem solving strategies.

Silversoft of London are converting their successful series of BBC Micro educational programs to run on the Electron.

They are also releasing what promises to be the first disassembler to be produced for the Electron.

Speedy loading on way

GOOD news for Electron owners frustrated with the slowness of saving and loading from cassette. Your problems may soon be over.

Two firms already well known in the BBC Micro world are planning ways of speeding things up.

The first is Pace of Bradford. Already one of the leading suppliers of disc filing systems for the BBC Micro, they are actively engaged in producing a similar system for the Electron.

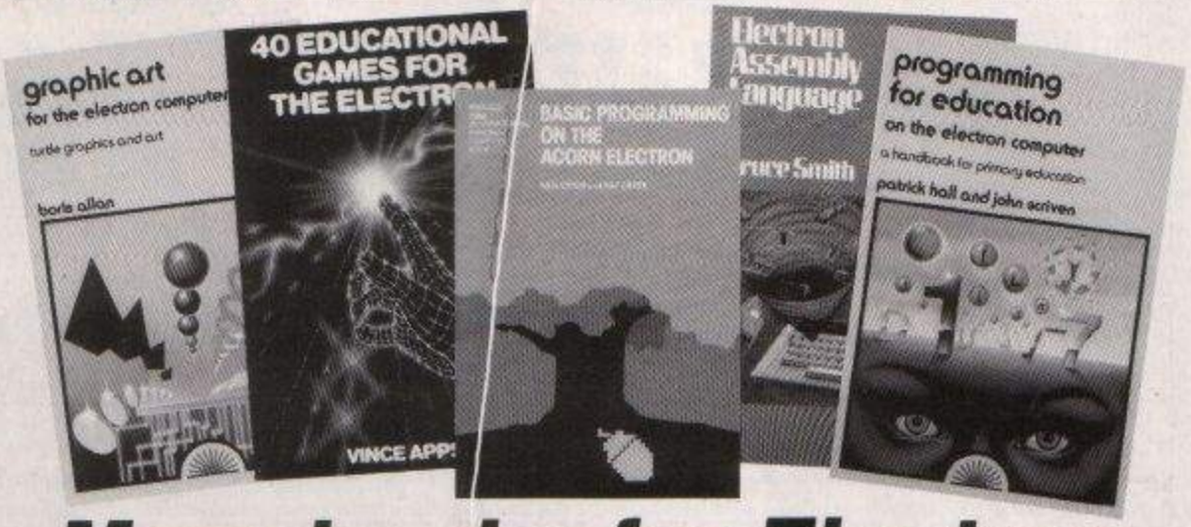
This means programs will be able to be loaded and saved in a matter of seconds rather than minutes, giving Electron users more time to use their machines.

From Ikon Computers of Dyfed comes the promise of another faster storage method, the Hobbit.

This is a tape based system whose speed approaches that of discs.

Well known to BBC Micro users, the Hobbit has recently had its price reduced and its speed increased.

This will make it a serious rival to disc based systems when it is released, hopefully later in the year.



More books for Electron

SPRING this year will see a flood of books covering all aspects of the Acorn Electron.

Beginners are well served by Neil and Pat Cryer's "Basic Programming on the Acorn Electron".

Well known for their book on the BBC Micro, the Cryers have repeated the same step by step, non-technical approach aimed at absolute novices.

However, the new books aren't all aimed at the elementary end of the market.

From Shiva comes Bruce Smith's Electron Assembly Language, a simple, well illustrated guide to using machine code to tap the hidden depths of the Electron.

With its treatment of the use of the Electron's built in assembler – one of its best features – and its explanation of the operating system, the book will open a whole new world to the Basic programmer.

Another specialist field, education, is well served by two of the new literary crop.

From Granada, who appear to be taking the lead in publishing for the Electron, comes "40 Educational games for the Electron" by Vince Apps.

Not to be outdone, Sunshine have brought out "Programming for Education on the Electron Computer".

Written by two teachers, Patrick Hall and John Scriven, the book is aimed at the primary education sector.

Sunshine have also produced "Graphic Art for the Electron Computer" by Boris Allan, the first book aimed specifically at exploring the Electron's graphics capabilities.

Database link planned

From Page 5

steps to ensure that Electron games now under development will be compatible with the new interface.

They have contacted all leading software houses giving details of the interface's software requirements and asking for their cooperation.

"We have been delighted by the help we've had from everyone", said Ray Threadgold of First Byte.

"Already A & F Software's Cylon Attack allows the use of our interface, and lots more are planned".

Other new products for the Electron are appearing thick and fast.

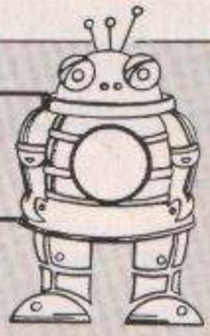
Not content with producing a joystick interface, Protek Computing of West Lothian has developed what promises to be the first modem for the Electron.

It allows users to talk to each other and mainframe computer databases such as Prestel over the telephone system.

This will vastly expand the scope of the micro.

Production is ready to go ahead as soon as British Telecom approves the production model.

Electron Eddie-torial



I WAS grabbed as soon as I walked in the door. "Pete, have a look at this, it's the first program I've ever written".

Tom put the cassette into the player and proudly LOADed his masterpiece.

"Watch this", he said as he typed in RUN and pressed Return.

It was really nice. Not the most original program I'd ever seen, but certainly an accomplished one.

He had made full use of the Electron's graphics abilities and the program was neat and crisp, well structured and well thought out. A competent piece of work.

"I like it", I said, wondering when I could use it in *Electron User*.

At that moment his dad came into the room.

"Look at that", Tom said, "it's my first program".

"Oh yes, very nice. What does it do?"

Tom's face dropped. What did it do?

I could have wept. I mean, did anyone walk up to Leonardo da Vinci when he'd finished the Mona Lisa and say: "Very nice, what does it do?"

What could I say? The guy had created a really nice program, showing that he had a thorough grasp of basic graphics and could use his knowledge practically.

Anyone who knew a little about micros would have been impressed, yet here he was, floored.

It was so frustrating. If he'd bought a radio instead of a micro and spoken to someone in Australia, everyone would have been thrilled.

If he'd have spent his money on a track suit and running shoes and trained up to run 26.2 miles in a marathon no one would bother asking why.

As it was he bought an Electron and used his time to understand how it worked and to create something that, however simple, was uniquely his.

He imagined it, thought about it and achieved it in practice. He'd used his micro creatively to express a part of himself.

And he'd been asked why.

I thought about it for quite a while, trying out different replies to his dad's question. Eventually I got the right answer.

So when you show someone your program and they ask you what it's for, don't bother trying to explain.

Just tell them: "If you have to ask the question, you'd never understand the reply".

Pete Bibby

Not so much what it does as how it does it . . .

Part Three of PETE BIBBY'S introduction to programming

Unravel that and pick up some hints

LAST month we saw how to write our own programs. Admittedly they were fairly trivial. But programs they were, exhibiting the basic features of any program.

This month we'll be looking at some ways of improving them and the output they produce on screen.

Again the examples won't be much to write home about, but it's the principles involved we're after.

Try the programs for yourself and see if you can understand how they work and if you can improve them.

Remember, it's a "hands on" course and you'll get a lot more out of it if you work through it on your Electron.

First though, let's have a look at what we've done so far.

We saw last month that a Basic program consists of a numbered sequence of instructions to the computer.

We entered these instructions, one after the other, giving each a line number.

These line numbers went up in steps of 10, allowing us to slip in other instructions if necessary.

We saw that we could replace a line with an altered version simply by typing in a new version, giving it the line number of the line we want it to replace.

If we wanted to get rid of a line completely we just typed in that line number and pressed Return.

We found that the Electron didn't obey these instructions straight away but waited until we typed in RUN, followed by the inevitable press of the Return key.

Finally we learned that we could use LIST to get the micro to display a list of instructions, NEW to clear it out of memory and CLS to clear the screen.

Now let's get cracking on the Electron. Type in Program I:

```
10 REM PROGRAM I
20 PRINT"HELLO"
30 PRINT"OUT"
40 PRINT"THERE"
```

Enter RUN and press Return. This will tell the micro to obey the instructions that it will find in its memory.

It starts at the one with the lowest line number. After that has been done it goes on to the next one and so on until it runs out of instructions.

As you'll see from the screen, the program displays the message:

```
HELLO
OUT
THERE
```

This is using the same techniques we came across last month. But the more observant of you might have noticed there is a new keyword.

This is the REM of line 10 and it is one of the easiest Basic statements to use and understand.

REM is short for remark, and the REM statement allows you to put remarks into your programs.

The Electron will ignore anything after a REM statement. When it finds one, it goes on to the next line number.

This allows you to put in your own remarks after the REM without upsetting the micro.

This can be very useful when you start to write longer programs. The remarks after the REM statements help to make the program more understandable.

Many a program has been saved from obscurity by a liberal use of REM statements.

In Program I the REM is used to make a note of the program title. The Electron doesn't read the PROGRAM I after the REM but goes straight to line 20.

I could have put in all sorts of remarks after the REM and the micro would still ignore them, no matter how personal I got!

Try leaving out the REM of line 10 and see what happens. The Electron is looking for a keyword, a Basic word of power. It is quite confused by the PROGRAM I which it finds after the line number.

Let's leave the REM statement for a while and go on to Program II, which prints out the same message in a different way.

But first, don't forget to type in NEW and press Return to get rid of the old program from memory.

```
10 REM PROGRAM II
20 PRINT"HELLO","OUT","THERE"
```

Some of you may have looked at Program I and wondered why I used three

string

PRINT commands in separate lines to print out the three bits of the message. Wouldn't one PRINT do?

Well, it will as Program II shows, though the message does look a bit spaced out.

The reason why it is spread across the screen is that we've put commas between the strings. *"The what?"* I hear you ask. The strings.

Put at its simplest, a string is just a piece of text placed in quotation marks. The Electron treats everything it finds inside quotation marks as one lump or string.

We've already used three strings in this article. They are "HELLO", "OUT" and "THERE". The Electron found one of these after each print statement of Program I.

The quotation marks told it that what followed was a string, and it printed out the whole string as one lump.

Notice that it doesn't print the quotation marks. They are just there to mark the beginning and the end of the strings.

Strings are very important in programming. But for the moment we'll leave it at that and go on to see why Program II printed the strings "HELLO", "OUT" and "THERE" in the way that it did.

As we might expect, the strings were displayed on the same line but without their inverted commas. But why were there the gaps between the words on the screen?

The answer is because we put commas between the strings – or we did if we typed the program in properly.

If we do this after a print statement it tells the micro to display each string on a separate part of the screen.

In the normal course of events the Electron divides the screen into four groups of 10 characters each. If instructed by commas between them, it will print the strings in separate fields.

Try:

PRINT "ONE", "TWO", "THREE", "FOUR"

and you'll see the separate print fields.

What happens if you enter:

PRINT "ONE", "TWO", "THREE", "FOUR", "FIVE", "SIX"

and press Return? Try it and see.

There's a lot more to these print fields, as they are called. But the point to grasp is that when commas separate the strings after a PRINT command then the strings are displayed in separate fields.

Let's see what happens when we run Program III:

```
10 REM PROGRAM III
20 PRINT"HELLO";"OUT";"THERE"
```

As you can see it's very much like Program II, only the commas have been changed to semicolons.

This effectively "glues" the strings together, overwriting the print fields we came across earlier.

The trouble is that the output looks a mess. There are no spaces between the words.

All the Electron does is print out the first string – "HELLO".

Then it finds the semicolon, which tells it to print whatever comes next straight away without any gaps.

The Electron doesn't know that you need spaces to make the words clear. If you want spaces, you have to add them yourself.

Run Program IV and see the result:

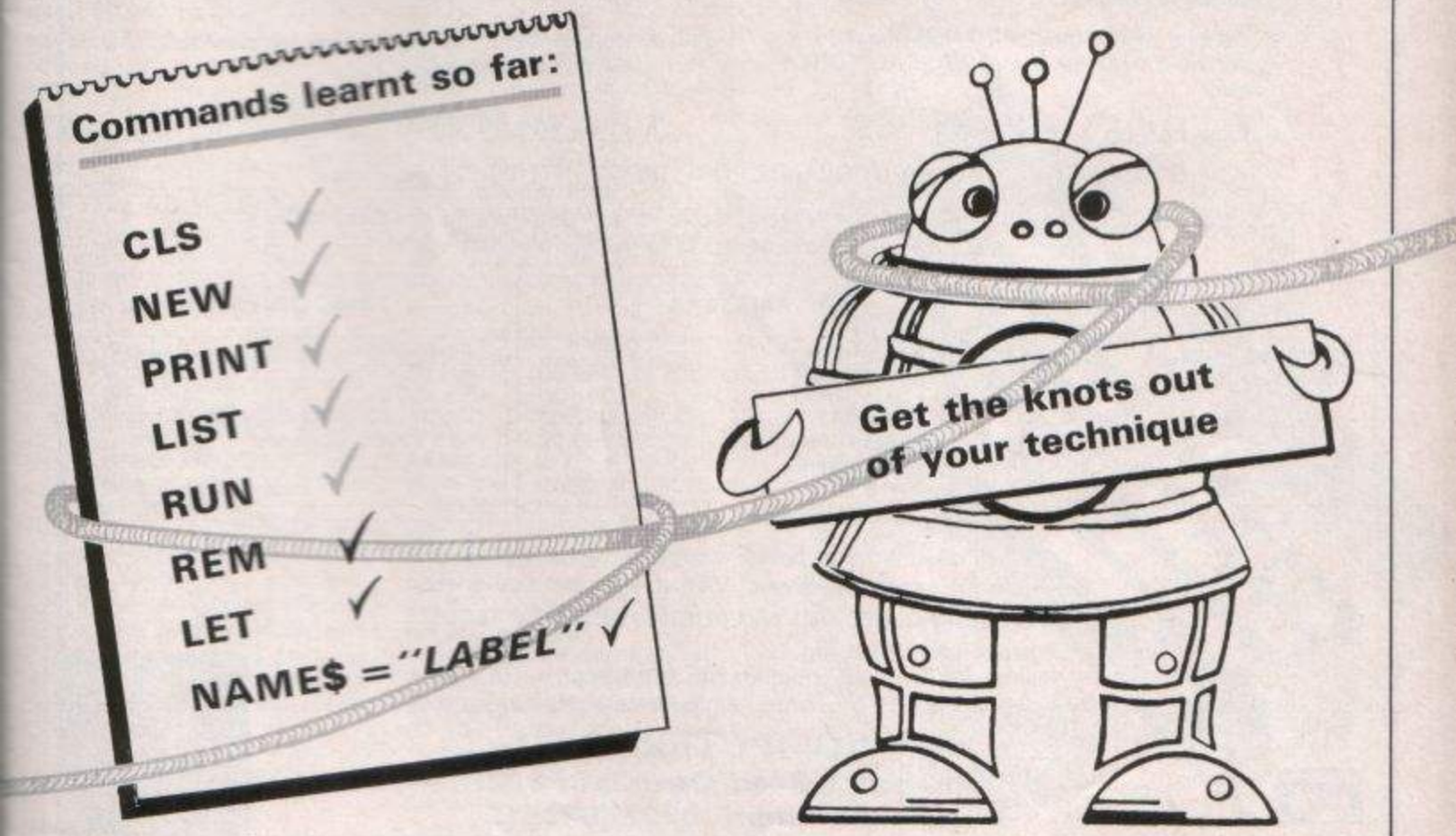
```
10 REM PROGRAM IV
20 PRINT"HELLO ";"OUT ";"THERE"
```

Here we've included the two necessary spaces in the strings. The Electron doesn't mind.

It will print out whatever it finds between the inverted commas – letters, numbers, spaces or any combination of them.

So now we've got our

→



From Page 9

program to print out the message on one line, neatly spaced. It's taken us a long time to get here, hasn't it?

Still, the principles involved will stand you in good stead in your programming career.

Mind you, we could have saved ourselves a lot of trouble if we'd run Program V:

```
10 REM PROGRAM V
20 PRINT"HELLO OUT THERE"
```

This just prints out one long string. Simple isn't it?

You may be wondering why we didn't do this in the first place. Well, with this message you could.

But the Electron sets a limit to the length of any one string. I leave you to work it out.

When you use long messages, you'll find that you need to know all the above techniques and how punctuation affects the PRINT command.

There's one more piece of punctuation that we haven't touched yet – the apostrophe.

Have a go at Program VI. Be careful when you type it in that you don't get confused between the puctuation marks:

```
10 REM PROGRAM VI
20 PRINT"HELLO "'"OUT "'"THERE"
```

We're back to the beginning again! Well, not quite, because we have done it in half the number of lines.

As you can see, the apostrophe between the strings tells the Electron to print the next string it finds at the beginning of a new line.

This can be quite useful for spacing out long messages.

Try using two or three apostrophes between the strings, and you'll see what I mean.

So we can now write out simple programs to display messages.

We're not just stuck with HELLO OUT THERE. We can put anything we want between the inverted commas and the Electron will display it.

The trouble is that the messages can get quite long. When you have had a little more experience you'll find that you're using PRINT to display quite large strings on the screen.

Take the case of the instructions for computer games. The part of the program that displays these uses exactly the same methods as we have done, only it has a lot more to say.

Also it might say the same thing at several points in the game, for example: "PRESS RETURN FOR ANOTHER GO".

It would be daft if we had to type in all the words every time we came to it.

Couldn't we give it a label and just tell the micro to print the label? It would save a lot of typing.

The answer is yes, and the use of labels is shown in Program VII:

```
10 REM PROGRAM VII
20 LET A$="HELLO "
30 LET B$="OUT "
40 LET C$="THERE"
50 PRINT A$
60 PRINT B$
70 PRINT C$
```

As you can see, the result is the same as before, only we've used a different method. We have given each of the strings a label.

Now when we want the Electron to do something with the string we can use the label to refer to it.

Since the label is shorter in length than the string, this saves a lot of typing.

The labels I have used are *A$, B$, C$*. The fact that they are in alphabetical order means nothing. I just picked them like that.

Nor does the name have to be so short – you can try other names.

The important thing to notice is that each one ends in a dollar sign, $. You'll find this above the 4 on the keyboard.

The rule is that if we want to refer to a string by a label – properly called a variable name – then that name must end in $ or else the Electron will get confused.

Let's take a closer look at Program VI. You'll notice that there is a new keyword in lines 20, 30 and 40.

This is the keyword LET. It tells the Electron that in future the string on the right of the equals sign will be referred to by the label on the other side of the equals sign.

It is important to remember that the label, the name you're giving to the string, comes after the LET.

The actual string you're labelling comes after the equals sign.

So lines 20, 30 and 40 assign labels to our three faithful old strings.

Lines 50, 60 and 70 then use PRINT to display the strings. But they refer to the strings by the labels we gave them in lines 20 to 40.

In this case using labels didn't save us much typing, but let's go back to the game instructions where it will.

It makes life much easier to have a line like:

```
10 LET MESSAGE$="PRESS RETURN
   FOR ANOTHER GO"
```

Now if you want the message you can just use the label in a line like:

```
40 PRINT MESSAGE$
```

rather than type in something like:

```
40 PRINT "PRESS RETURN
   FOR ANOTHER GO"
```

which would be fairly time consuming if we wanted the same message over and over again.

You'll see from Program VIII that we can use the labels exactly as if they were the strings themselves.

Here we only use one PRINT command to display the message, with the punctuation between the labels acting just as if the string themselves were there.

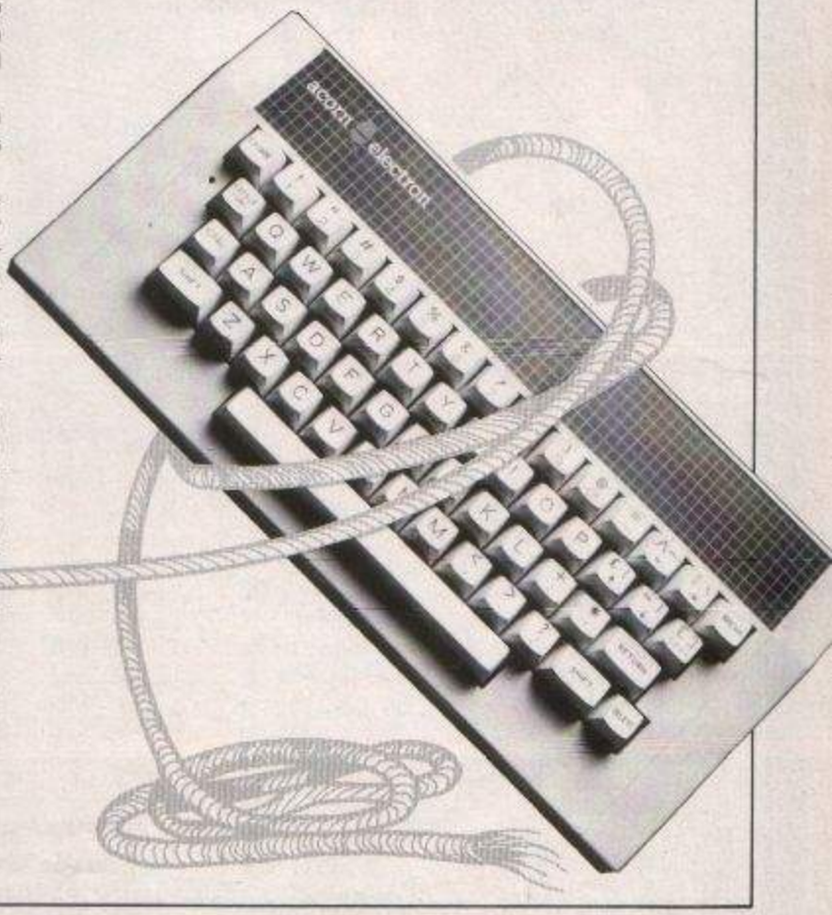
```
10 REM PROGRAM VIII
20 LET A$="HELLO "
30 LET B$="OUT "
40 LET C$="THERE"
50 PRINT A$;B$;C$
```

Try it out with commas and apostrophes between the labels and see for yourself what happens.

There's a lot more to strings than we have covered in this article, but for the moment that's enough.

Try writing a few of your own programs to print messages on the screen.

Use labels as much as possible to make your life easier, and soon strings will become second nature.



Notebook Part 3

GRID is a program that gets your Electron to display a multicoloured series of horizontal and vertical lines. Once these lines are drawn there's a short delay and then the lines start changing colour randomly.

Type it in and see if you can understand how it works. If you want to see some flashing lines then change the RND(7) in lines 60, 110 and 170 to RND(15).

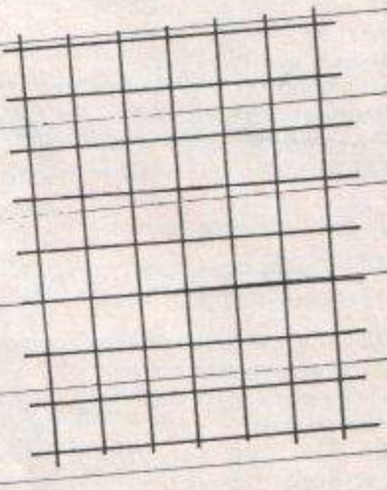
```
10 REM GRID
20 REM (C) ELECTRON USER
30 MODE 2
40 VDU 23,1,0;0;0;0;
50 FOR X = 0 TO 1279
     STEP 64
60 GCOL 0,RND(7)
70 MOVE X,0
80 DRAW X,991
90 NEXT X
100 FOR Y= 0 TO 1023
     STEP 32
110 GCOL 0,RND(7)
120 MOVE 0,Y
130 DRAW 1215,Y
140 NEXT Y
150 REPEAT
160 VDU 19,RND(7),RND(7)
     ,0,0,0
170 FOR P=1 TO 500
     :NEXT P
180 UNTIL FALSE
```

REM statements

FOR...NEXT lines 50-90

Lines 100 to 140 FOR...NEXT loop

Lines 150 to 180 REPEAT ... UNTIL loop



10,20 REM statements to give information.

30 Puts the Electron in Mode 2.

40 Switches off the flashing cursor.

50-90 Form a FOR... NEXT loop that puts the vertical lines on the screen.

60,110 Pick random colours for the Electron to draw the lines in. STEP 64 just decides the gap between each line. Try different values and you will see what happens.

70 Starts each line at the bottom of the screen.

80 Draws the line to the top of the screen.

100-140 Form a FOR... NEXT loop which draws lines across the screen.

120 Starts each horizontal line at the left of the screen, each one slightly higher than the other.

130 Draws the line across the screen.

150-180 Form a REPEAT... UNTIL loop which, as the condition is FALSE, carries on forever.

160 The VDU19 changes the colour of the lines, randomly.

170 Forms a FOR... NEXT loop which produces a delay between the swapping of the coloured lines.

Examples of punctuation in the line numbers on the left:

50-70 Means all the line numbers inclusive.

50,70 Means just the two numbers, 50 and 70.

Trevor Roberts

Make light work of listings!

All program listings in *Electron User* have been put on tape – to save you the chore of keying them in yourself. Four tapes are now available – for the February, March and April issues, plus a bumper tape of all the programs from the first few introductory issues.

On the April tape:

SPACEHIKE A hopping arcade classic. **FRIEZE** Electron wallpaper. **PELICAN** Cross roads safely. **CHESSTIMER** Clock your moves. **ASTEROID** Space is a minefield. **LIMERICK** Automatic rhymes. **ROMAN** Numbers in the ancient way. **BUNNYBLITZ** The Easter program. **DOGDUCK** The classic logic game. **NOTEBOOK** Coloured grids. **BINARY** A base program.

On the March tape:

CHICKEN Let dangerous drivers test your nerve. **COFFEE** A tantalising word game from Down Under. **PARKY'S PERIL** Parky's lost in an invisible maze. **REACTION TIMER** How fast are you? **BRAINTEASER** A puzzling program. **COUNTER** Mental arithemetic can be fun! **PAPER, SCISSORS, STONE** Out-guess your Electron. **CHARACTER GENERATOR** Create shapes with this utility. **FUNNY POLYGONS** Fast graphics going round in circles. **RABBITS** Easter bunnies all over! **DRAW** Multi-coloured lines. **MEAN** Just an average program.

On the February tape:

NUMBER BALANCE Test your powers of mental arithmetic. **CALCULATOR** Make your Electron a calculator. **DOILIES** Multi-coloured patterns galore. **TOWERS OF HANOI** The age old puzzle. **LUNAR LANDER** Test your skill as an astronaut. **POSITRON INVADERS** A version of the old arcade favourite. **MOON RESCUE** Avoid the asteroids and save the spacemen. **STARS** A program making pretty pictures. **TAPESTRY** Symmetry and colour combine.

On the introductory tape:

ANAGRAM Sort out the jumbled letters. **DOODLE** Multicoloured graphics. **EUROMAP** Test your geography. **KALEIDOSCOPE** Electron graphics run riot. **CAPITALS** New upper case letters. **ROCKET, WHEEL, CANDLE** Three fireworks programs. **BOMBER** Drop the bombs before you crash. **DUCK** Simple animation. **METEORS** Collisions in space. **COMBINATIONS** Crack the hidden code. **BUZZ WORD GENERATOR** Let the Electron help you impress. **SIMON** Reactions and memory put to the test. **3-D PLOT** Enter a new dimension. **PLUS LOTS MORE!**

HOW TO ORDER

Please send me the following *Electron User* cassette tapes:

Eleven programs from the April issue ..... £

Twelve programs from the March issue ..... £

Nine programs from the February issue ..... £

26 programs from the introductory issues ..... £ _____

I enclose the sum of £ _____

Name .....

Address .....

.....

POST TO: Tape Offer, Electron User, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.

ONLY £3.75 each

A joint presentation by
The Micro User & Electron User

Don't miss the great new for the Electron

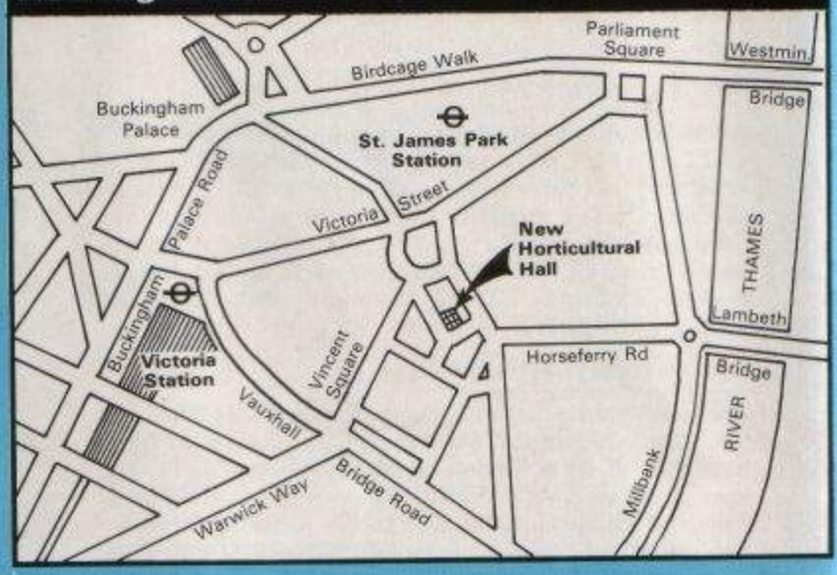
Here's your big chance to catch up on all that's been happening recently in the fast-developing world of the Electron and BBC Micro. And there's so much new to excite and intrigue you . . .

NEW programs from the fertile minds of Britain's leading software writers – games galore, plus a growing number of new packages for teachers and for industrial and business users.

NEW hardware add-ons that expand even more the power and versatility of your micro.

Electronics wizards regard both the Electron and the BBC Micro as a challenge to their ingenuity. Their latest creations on display at the Electron and BBC Micro User Show will astound and delight you!

How to get there



VOUCHER WORTH £1

This voucher is worth £1 per person off the normal admission price of £3 (adults) and £2 (children)
(Valid for a maximum of 4 people)

Electron & BBC Micro User Show

10am - 6pm, Thursday, 29 March
10am - 6pm, Friday, 30 March
10am - 6pm, Saturday, 31 March
10am - 4pm, Sunday, 1 April

New Horticultural Hall
Greycoat Street, London SW1

School and College Groups

Entry only £1 per student if bookings are made in advance. Send your cheque (made payable to Database Publications) and SAE to:

Electron & BBC Micro User Show
68 Chester Road, Hazel Grove
Stockport SK7 5NY
Tel: 061-456 8383

Spring show of all that's and BBC Micro

New Horticultural Hall

(Westminster Exhibition Centre)

Thursday to Sunday, March 29 to April 1

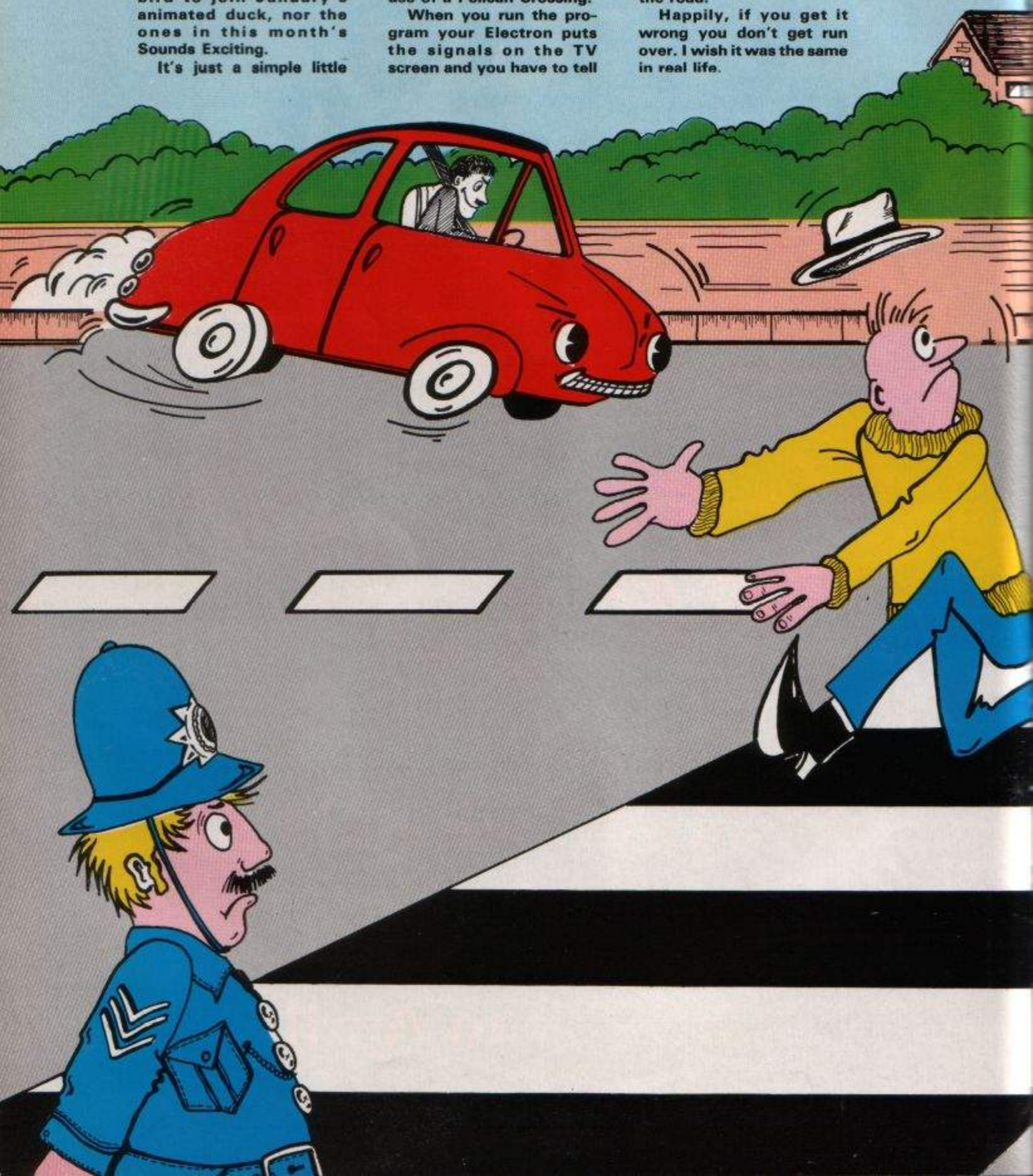
Pelican

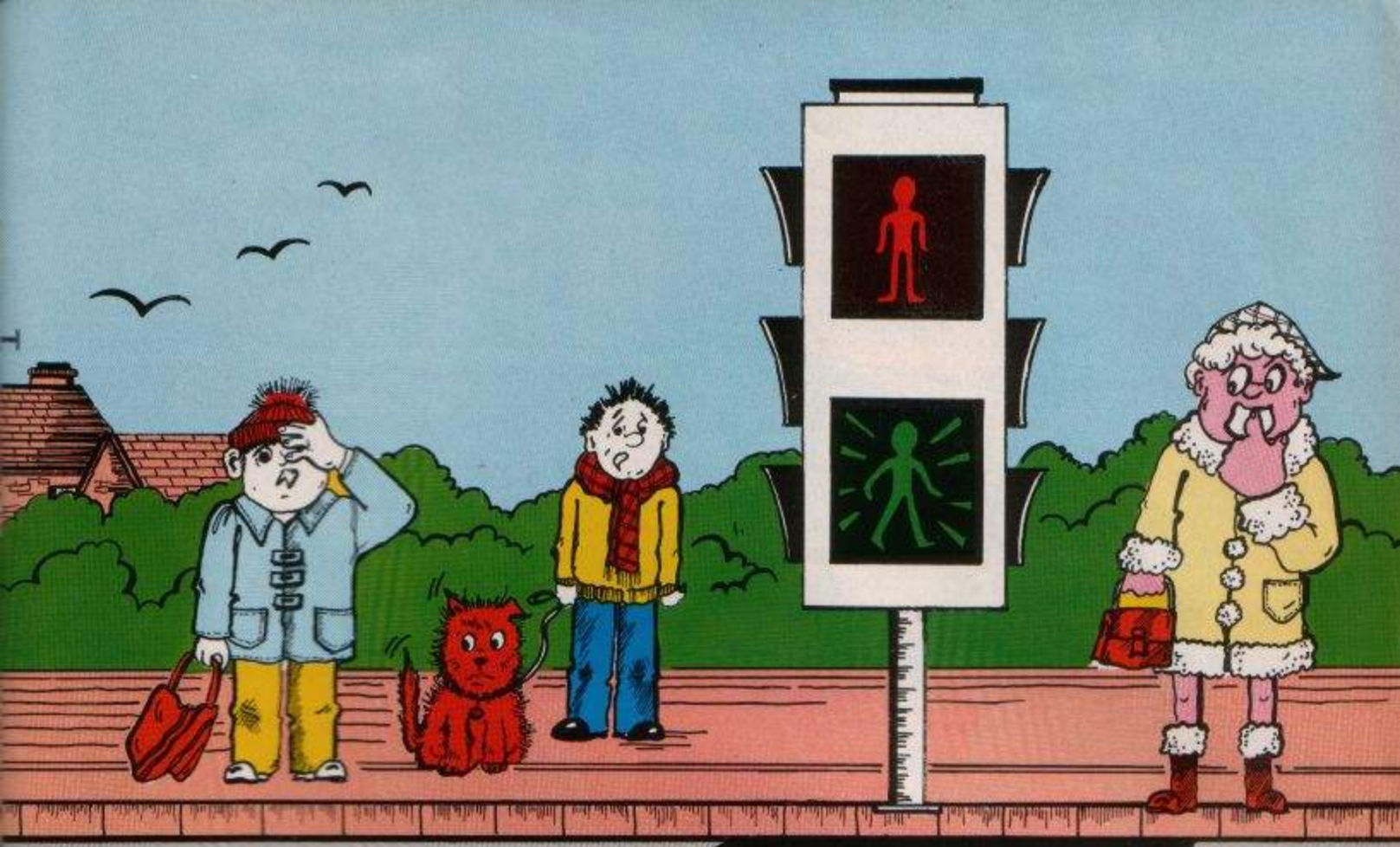
NO, Pelican isn't another bird to join January's animated duck, nor the ones in this month's Sounds Exciting.

It's just a simple little program that simulates the use of a Pelican Crossing.

When you run the program your Electron puts the signals on the TV screen and you have to tell it when it's safe to cross the road.

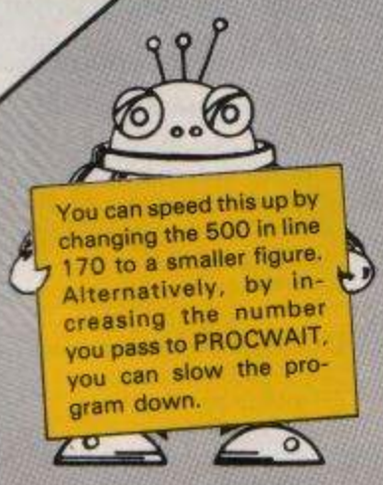
Happily, if you get it wrong you don't get run over. I wish it was the same in real life.





By Alan McLachlan

```
 10 REM BY ALAN M.
 20 REM (C) ELECTRON USER
 30 MODE 2
 40 VDU 23;8202;0;0;0;
 50 ON ERROR GOTO 1810
 60 PROCINST
 70 PROCSCREEN
 80 PROCCHARS
 90 REPEAT
100 PROCINIT
110 key_flag=FALSE
120 PROCPEDLIGHT(780,470
    ,"WAIT",8)
130 PROCGREEN
    :REM GO FOR TRAFFIC
140 A$=GET$
    :IF A$() "C"
    THEN 140
150 PROCWAIT(500,1)
160 PROCAMB
    :REM STOP FOR TRAFFIC
170 PROCWAIT(500,1)
180 PROCRED
190 PROCPIPS
200 PROCFLAMB
210 UNTIL FALSE
220 REM **********
    **********
230 DEF PROCINIT
    :REM SET UP SCREEN
240 VDU 4,28,0,31,19,27
    :COLOUR 135
    :CLS
250 VDU 5
260 GCOL 0,4
270 MOVE 100,950
    :PRINT "TRAFFIC"
280 MOVE 160,900
    :PRINT "LIGHT"
290 GCOL 0,1
300 MOVE 0,96
    :PRINT "PRESS X WHEN"
310 MOVE 0,50
    :PRINT "SAFE TO CROSS"
320 GCOL 0,0
330 MOVE 700,350
    :PRINT "PRESS C"
340 MOVE 690,310
    :PRINT "TO STOP"
350 MOVE 690,270
    :PRINT "TRAFFIC"
360 ENDPROC
370 DEF PROCSCREEN
380 CLG
    :VDU 5
390 PROCBOX1
```



You'll be ITCHING to get your hands on the funniest program of 1984!

Fleas!
Bed Bugs!
And all you wanted was a quiet night . . .

The pests are after your feet and you'll have to move fast to stop them. Swot them with a jam sandwich or crunch them with your false teeth.

If you're desperate you can always phone for help. But whatever you do, do it quickly. You need cunning tactics and nimble fingers!

Bed Bugs guarantees hours of hilarity for the whole family.

Obtainable from your Acorn dealer or send in the coupon below

ORDER FORM

Please send me BED BUGS:

- ☐ *BBC 'B' cassette – £6.95*
- ☐ *Electron cassette – £6.95*
- ☐ *BBC 40-track disc – £8.95*
- ☐ *BBC 80-track disc – £8.95*

Add 50p p&p (post free 2 or more)

Name .....

Address .....

.....

.....

.....

☐ I enclose cheque made payable to Optima Software Ltd.

☐ I wish to pay by ☐ Access ☐ Visa

No .....

Signed .....

Expiry date .....

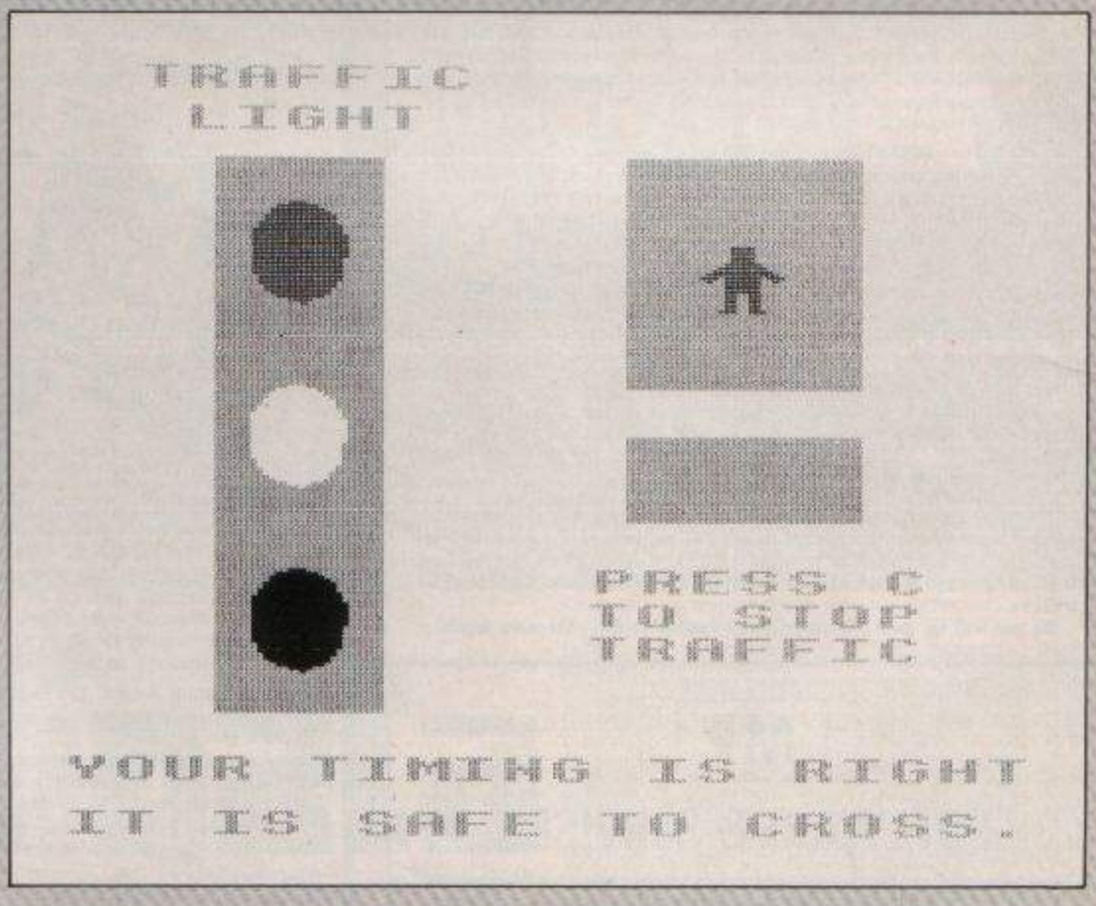
Optima Software Ltd., 36 St. Petersgate, Stockport SK1 1HL.

OPTIMA SOFTWARE

From Page 17

This listing was produced using a special formatter which breaks one program line over several lines of listing. When entering a line don't press Return until you come to the next line number. Full details of the formatter are given on Page 4 of the February issue.

```
400 PROCBOX2
410 PROCBOX3
420 PROCPEDMAN(13)
430 PROCLIGHTON(310,730
    ,1)
440 PROCLIGHTON(310,510
    ,3)
450 PROCLIGHTON(310,290
    ,2)
460 ENDPROC
470 REM **********************
    ************
480 DEF PROCLIGHTON(X%,Y%
    ,C%)
490 VDU 19,1,0;0;
500 VDU 19,2,0;0;
510 VDU 19,3,0;0;
520 VDU 19,13,1;0;
530 R%=60
540 VDU 29,X%;Y%;
550 GCOL 0,C%
560 MOVE 0,0
570 FOR I=0 TO PI *3
    STEP .25
580 MOVE 0,0
590 PLOT 85,R%*COS I,R%*
    SIN I
600 NEXT
610 VDU 29,0;0;
620 ENDPROC
630 REM **********************
    ************
640 DEF PROCRED
650 IF key_flag ENDPROC
660 VDU 19,1,1;0;
670 VDU 19,2,0;0;
680 VDU 19,3,0;0;
690 VDU 19,13,2;0;
700 PROCBOX3
710 ENDPROC
720 REM **********************
    ************
730 DEF PROCAMB
740 IF key_flag ENDPROC
750 VDU 19,3,3;0;
760 VDU 19,2,0;0;
770 VDU 19,1,0;0;
780 ENDPROC
790 REM **********************
    ************
800 DEF PROCFLAMB
810 IF key_flag ENDPROC
820 VDU 19,1,0;0;
830 FOR I%=0 TO 10
840 VDU 19,3,3;0;
850 VDU 19,13,2;0;
860 PROCWAIT(90,1)
870 VDU 19,13,0;0;
880 VDU 19,3,0;0;
890 PROCWAIT(90,1)
900 NEXT
910 ENDPROC
920 REM **********************
    ************
930 DEF PROCGREEN
940 VDU 19,2,2;0;
950 VDU 19,3,0;0;
960 VDU 19,1,0;0;
970 VDU 19,13,1;0;
980 ENDPROC
990 REM **********************
    ************
1000 DEF PROCPEDLIGHT(A%
    ,B%,L$,C%)
1010 GCOL 0,C%
1020 MOVE A%,B%
1030 PRINT L$
1040 ENDPROC
1050 REM **********************
    ************
1060 DEF PROCPEDMAN(C%)
1070 PROCCHARS
1080 GCOL 0,C%
1090 MOVE 870,750
1100 VDU 224,10,8,8,228,225
    ,227,10,8,8,226
1110 ENDPROC
1120 REM **********************
    ************
1130 DEF PROCCHARS
1140 VDU 23,224,0,0,0,60
    ,60,60,60,60
1150 VDU 23,225,126,255,255
    ,255,126,126,126,126
1160 VDU 23,226,126,102,102
    ,102,102,102,231,0
1170 VDU 23,227,0,0,128,192
    ,224,64,0,0
1180 VDU 23,228,0,0,1,3,7
    ,2,0,0
1190 VDU 23,255,255,255,255
    ,255,255,255,255,255
1200 ENDPROC
1210 REM **********************
    ************
1220 DEF PROCBOX1
1230 VDU 7
1240 GCOL 0,0
1250 MOVE 200,180
    :MOVE 200,840
    :PLOT 85,420,840
1260 MOVE 200,180
    :MOVE 420,180
    :PLOT 85,420,840
1270 ENDPROC
1280 REM **********************
    ************
1290 DEF PROCBOX2
1300 GCOL 0,0
1310 MOVE 750,570
    :MOVE 750,840
    :PLOT 85,1050,840
1320 MOVE 750,570
    :MOVE 1050,570
    :PLOT 85,1050,840
1330 ENDPROC
1340 REM **********************
    ************
1350 DEF PROCBOX3
1360 GCOL 0,0
1370 MOVE 750,410
    :MOVE 750,510
    :PLOT 85,1050,510
```



Pelican listing

From Page 19

```
1380 MOVE 750,410
     :MOVE 1050,410
     :PLOT 85,1050,510
1390 ENDPROC
1400 REM **********************
     ************
1410 DEF PROCPIPS
1420 P=0
     :REPEAT P=P+1
1430 SOUND 1,-15,200,1
1440 PROCCHECKX
1450 UNTIL P=25 OR key_flag=
     TRUE
1460 ENDPROC
1470 REM **********************
     ************
1480 DEF PROCINST
1490 COLOUR 135
     :GCOL 0,135
     :GCOL 0,0
     :CLS
     :CLG
1500 VDU 5
1510 MOVE 130,900
1520 PRINT "PELICAN CROSSING"
1530 MOVE 130,860
1540 PRINT "****************"
1550 VDU 4
1560 COLOUR 1
1570 PRINT TAB(1,10)"PRESS
     THE 'C' KEY
                    WHEN YOU
     ARE READY
                 TO STOP TRAFF
     IC"
1580 COLOUR 4
1590 PRINT TAB(1,20)"PRESS
     THE 'X' KEY
                    WHEN IT
     IS SAFE
                 TO CROSS
     THE ROAD."
1600 PRINT TAB(2,29)"ANY KEY
     TO START";
1610 A$=GET$
1620 ENDPROC
1630 REM **********************
     ************
1640 DEF PROCCHECKX
1650 TIME =0
     :*FX15,1
1660 REPEAT
1670 A$=INKEY$ (0)
1680 UNTIL TIME >=30 OR A$<>
     ""
1690 IF A$<>"X" AND A$<>""
     THEN VDU 4
     :CLS
     :COLOUR 0
     :PRINT '"THE 'X' KEY
      I SAID"
     :PROCWAIT(900,0)
     ELSE IF A$="X"
     THEN VDU 4
     :CLS
     :PRINT '"YOUR TIMING
      IS RIGHT"'"IT IS SAFE
      TO CROSS.";
     :PROCWAIT(900,0)
1700 IF A$=""
     THEN VDU 4
     :CLS
     ELSE key_flag=TRUE
1710 VDU 5
1720 ENDPROC
1730 REM **********************
     ************
1740 DEF PROCWAIT(W%,G%)
1750 IF key_flag ENDPROC
1760 DL=0
     :REPEAT DL=DL+1
1770 IF INKEY (-67) AND G%
      AND NOT key_flag
     VDU 4
     :CLS
     :PRINT "YOU BLEW IT"
     :VDU 5
     :key_flag=TRUE
     :FOR DL2=0 TO 200
     :NEXT
1780 UNTIL DL=W% OR key_flag
1790 ENDPROC
1800 REM **********************
     ************
1810 MODE 7
1820 REPORT
     :PRINT " In line ";
     ERL
```

This listing is included in this month's cassette tape offer. See order form on Page 43.

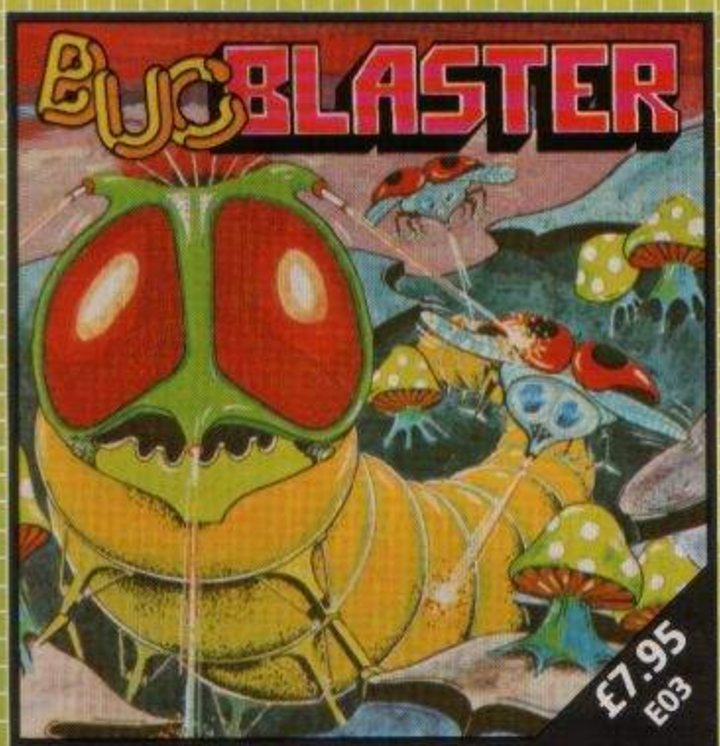
THE GAME TO MAKE YOUR SKIN CRAWL

BUGBLASTER

THE ACTION PACKED HIT REWRITTEN FOR ELECTRON

Alligata presents a superb range of software products that are designed specially for you. Games that cleverly combine full machine code and high resolution, full colour graphics to create hours of fun and excitement. And utilities that have been developed to open new doors and help get the best from your Electron micro. If it's to be outstanding quality and amazing value for money then Alligata has to be your choice.

Send a stamped addressed envelope for our full colour catalogue which gives details of the complete range.



A superb action packed arcade special. A really fast implementation of the splendid 'centipede.' Features include spiders, mushrooms, centipedes and the mushroom poisoning scorpion affectionately known as 'Brian.' The better you get the faster the action. Nerve tingling excitement should keep you up all night!

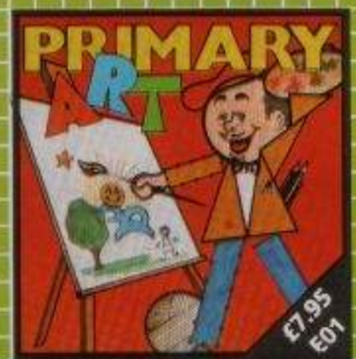
Experience all the speed and excitement of the arcade spectacular

WRITE OR PHONE YOUR ORDER TODAY!

also available from all good software stockists.

E05 Scribe II £9.95
Produce professional letters and documents, speedily and easily, with this superb word processing program – handling up to 2 A4 pages as one file. Simple to use, yet very powerful, Scribe II handles up to 600 lines of text with 80 characters per line screen display. Compatible with most printers.

Land your moon buggy and rescue a precious cargo, destroying all opposition on the way; finding your way back to the mother ship start again against greater odds.



Create a picture to be proud of – place pre-programed shapes in any position, any size or any colour. Features free-hand drawing and animation effects.

The fun-filled way to learn to tell the time. The combination of sound and simple display complement each other to produce a very easy to use teaching package.

Keeping your money in your pocket enjoy all the excitement of beating the one arm bandit.

Alligata

Alligata Software Ltd. 178 West Street, Sheffield S1 4ET. Tel: (0742) 755005

Despatch is normally made on receipt of order and should reach you within 7 days.

INDICATE PROGRAMS REQUIRED
E03☐ E06☐ E01☐ E02☐ E04☐ E05☐

I enclose cheque/PO* for £* _____ Charge my Access/Visa £* _____

Card No. _____ Signature _____

Name _____

Address _____

*payable to Superior Systems Ltd., 178 West Street, Sheffield S1 4ET.
*allow 75p for post and packaging.

BOOKSHELF

Limerick illustration from Take off with the Electron and BBC Micro

IF you're looking for just another book of games listings then "Take Off with the Electron and BBC Micro" will probably disappoint you.

However if you're after 11 interesting programs with lots of ideas on how to improve them then this is for you.

For listings are not just programs in their own right but can be expanded. And the book tells you how to do this.

Take off with the Electron and BBC Micro
Granada Publishing

It starts with a concise but thorough description of elementary Basic and then goes on to the listings.

Each program has a chapter to itself and all chapters have the same structure.

You first read a description of what the listing does. Then comes the listing itself.

These are easy to read and the authors claim that "it is very unlikely that there are any mistakes in the listings". Brave words and, as far as I can tell, true ones.

The listings are useful and fun. But the real value of the book, to my mind anyway, is in what follows them.

Each chapter has a well-annotated flow chart illustrating how it works. Then comes a line-by-line description of the program, very much like the ones you'll find in *Electron User*.

When you've read how the program works there's a discussion of the keywords involved, nicely cross-referenced to the other listings.

Then comes a section describing one of the techniques used in the program, such as user defined characters and file handling.

These really add to the book's value.

Finally you reach the "Take off from here" section. This gives suggestions about modifying and improving the programs given.

I like the book. For the person who's taken his first faltering steps in Basic and would like to start more ambitious programming it's excellent.

The authors strike just the right level, not too difficult, not too simple, while keeping it all interesting.

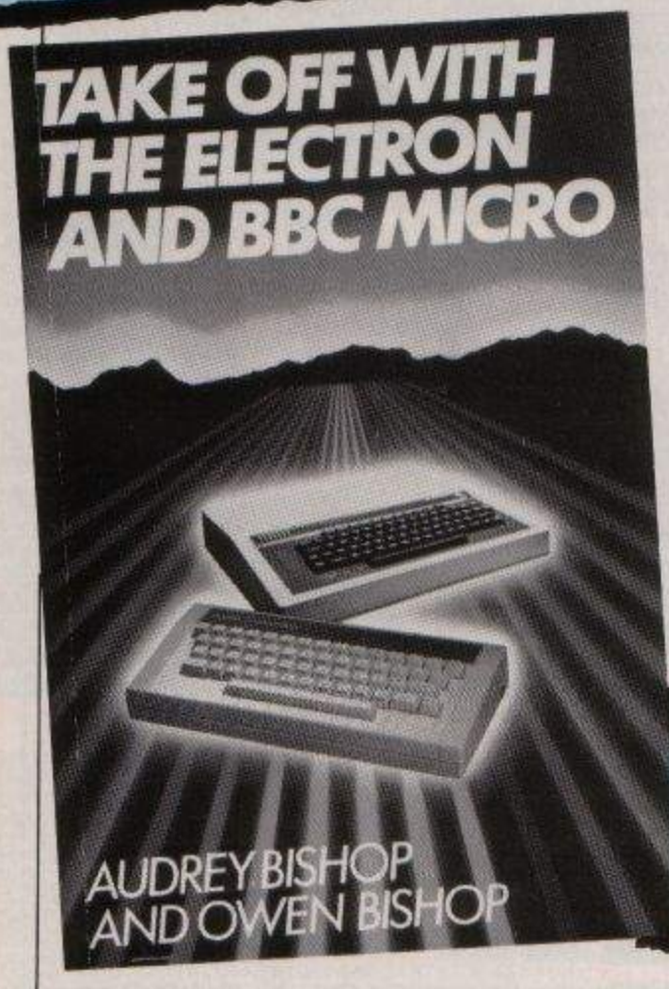
Also the programs are nicely chosen. They range from the limerick writer (reproduced here) to a stunt car game via a music maker and a weather forecasting program.

All are short and easy to experiment with, and all of them are well explained.

Thoroughly recommended.

Nigel Peters

```
There was a fat writer called Andrew
Who seldom said "Yes" and then "Can do"
And went out one day
To romp in the hay
That hopeless fat writer called Andrew
PRESS ANY KEY TO CONTINUE
```




```
There was a poor toddler called Sarah
Who seldom ate steak so much rarer
And went out one night
To put out the light
That hopeless poor toddler called Sarah
PRESS ANY KEY TO CONTINUE
```

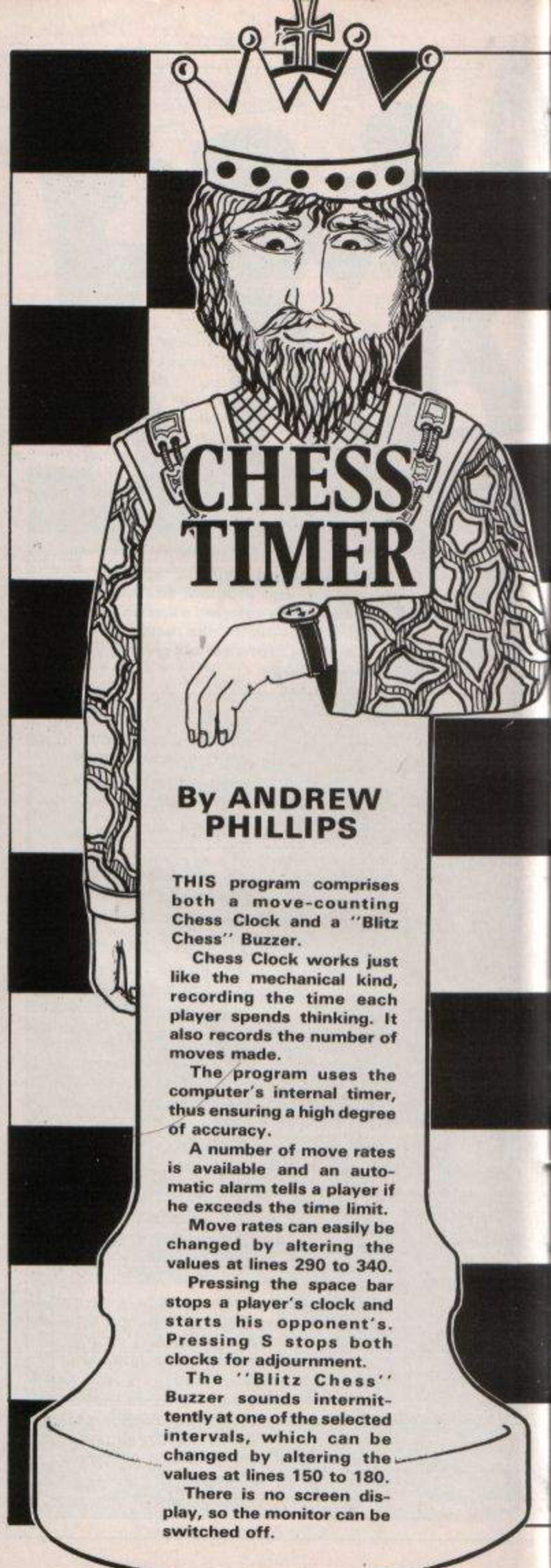
Limerick listing

```
 1 REM FROM TAKE OFF WITH
 2 REM THE ELECTRON AND
 3 REM BBC MICRO
 4 REM BY OWEN AND
 5 REM AUDREY BISHOP
 6 REM GRANADA PUBLISHING
 7 REM PRICE £5.95
 8 REM USED WITH THANKS
10 REM **LIMERICK**
20 MODE 4
30 READ A
   :DIM A$(A)
40 FOR J=1 TO A
   : READ A$(J)
   :NEXT J
50 READ B
   :DIM B$(B)
60 FOR J=1 TO B
   : READ B$(J)
   :NEXT J
70 READ C
   :READ D
   :DIM C$(C), D$(D,C)
80 FOR K=1 TO C
   : READ C$(K)
90 FOR J=1 TO D
   : READ D$(J,K)
   :NEXT
   :NEXT
100 READ E
   :READ F
   :DIM E$(E), F$(F,E)
110 FOR K=1 TO E
   : READ E$(K)
120 FOR J=1 TO F
   : READ F$(J,K)
   :NEXT
   :NEXT
130 READ G
   :DIM G$(G)
140 FOR J=1 TO G
   : READ G$(J)
```

This listing was produced using a special formatter which breaks one program line over several lines of listing. When entering a line don't press Return until you come to the next line number. Full details of the formatter are given on Page 4 of the February issue.

```
   :NEXT J
50 READ H
   :DIM H$(H)
60 FOR J=1 TO H
   : READ H$(J)
   :NEXT J
70 READ I
   :DIM I$(I)
80 FOR J=1 TO I
   : READ I$(J)
   :NEXT J
90 REPEAT
00 CLS
10 PRINT TAB(16,5)"LIMERICK"
20 RA=RND(A)
   :RB=RND(B)
   :RC=RND(C)
   :RE=RND(E)
30 PRINT TAB(0,10)"There
    was a "A$(RA)" "B$(RB)
   " called "C$(RC)
40 PRINT "Who "H$(RND(H))
   " "D$(RND(D),RC)
50 PRINT "And "I$(RND(I))
   " "E$(RE)
60 PRINT "To "F$(RND(F)
   ,RE)
70 PRINT "That "G$(RND(G))
   " "A$(RA)" "B$(RB)" calle
   d "C$(RC)
80 PRINT TAB(8,30)"PRESS
    ANY KEY TO CONTINUE"
90 key$=GET$
80 UNTIL FALSE
10 DATA 3, poor, fat, rich
320 DATA 4, singer, writer
    , toddler, pop star
330 DATA 3,2, Andrew,liked
    eating cheese fondue
340 DATA said "Yes"and then
    "Can do"
350 DATA Sarah, ate steak
    so much rarer
360 DATA thought no one
    would dare 'er
370 DATA Winny, liked wearing
    a pinny
380 DATA looked horribly
    skinny
390 DATA 2,3, one day, romp
    in the hay
400 DATA join in the fray
410 DATA go out to play
420 DATA one night, have
    a good fight
430 DATA put out the light
440 DATA just be polite
450 DATA 2, silly, hopeless
460 DATA 2, always, seldom
470 DATA 2, went out, started
```

LIMERICK is one of 11 listings in "Take Off With The Electron and BBC Micro" by Audrey and Owen Bishop. It is published by Granada Publishing, price £5.95. Our thanks to Granada for permission to reproduce the game.



CHESS TIMER

By ANDREW PHILLIPS

THIS program comprises both a move-counting Chess Clock and a "Blitz Chess" Buzzer.

Chess Clock works just like the mechanical kind, recording the time each player spends thinking. It also records the number of moves made.

The program uses the computer's internal timer, thus ensuring a high degree of accuracy.

A number of move rates is available and an automatic alarm tells a player if he exceeds the time limit.

Move rates can easily be changed by altering the values at lines 290 to 340.

Pressing the space bar stops a player's clock and starts his opponent's. Pressing S stops both clocks for adjournment.

The "Blitz Chess" Buzzer sounds intermittently at one of the selected intervals, which can be changed by altering the values at lines 150 to 180.

There is no screen display, so the monitor can be switched off.

This listing was produced using a special formatter which breaks one program line over several lines of listing. When entering a line don't press Return until you come to the next line number. Full details of the formatter are given on Page 4 of the February issue.

```
 10 REM "CHESS TIMER"
 20 REM A.Phillips
 30 REM (C) ELECTRON USER
 40 ON ERROR GOTO 470
 50 MODE 5
    :VDU 23;8202;0;0;0;
 60 VDU 19,2,4;0;17,130
    ,12
 70 *FX11
 80 *FX4,1
 90 COLOUR 1
    :PRINT TAB(4,3)"CHESS
     TIMER"TAB(4)STRING$(11
    ,"_")
    :COLOUR 3
100 PRINT TAB(2,8)"SELECT
     FUNCTION:"'''" 1
     - Chess Clock"'''
    " 2 - 'Blitz Chess'"''
    "     Buzzer"
110 ON INSTR("12",GET$ )
     GOTO 220 ,120
    ELSE 110
120 CLS
    :COLOUR 1
    :PRINT TAB(0,2)"'BLITZ
     CHESS' BUZZER"'
    STRING$(20,"_")
    :COLOUR 3
130 PRINT TAB(2,8)"SELECT
     INTERVAL:"'''" 1
     - 5 seconds"'''" 2
     - 10 seconds"'''"
     3 - 15 seconds"'''
    " 4 - 20 seconds"
140 ON INSTR("1234",
    GET$ ) GOTO 150 ,160
    ,170 ,180
    ELSE 140
150 Interval%=5
    :GOTO 190
160 Interval%=10
    :GOTO 190
170 Interval%=15
    :GOTO 190
180 Interval%=20
190 CLS
    :PRINT TAB(3,8)"Switch
     off TV"''"then press
     SPACE BAR"'"  to start
     buzzer"
200 PRINT TAB(1,25)"(Buzzer
     Interval ="''SPC (4);I
    nterval%;" seconds)"
210 REPEAT UNTIL GET$ =
    " "
    :PRINT TAB(0,8)
    SPC (100)
    :PROCbuzz(Interval%)
220 CLS
    :@%=2
    :Rate%=0
    :All%=0
230 DIM T%(1),C%(1),S%(1)
    ,M%(1),H%(1),Move%(1)
240 ENVELOPE 1,0,0,0,0
    ,0,0,0,126,-4,0,-1
    ,126,100
250 COLOUR 1
    :PRINT TAB(4,2)"CHESS
     CLOCK"TAB(4)STRING$(11
    ,"_")
    :COLOUR 3

260 PRINT TAB(2,7)"SELECT
     MOVE RATE:"'''" 0
     - No limit"'''" 1 -
     20 moves/hour"'''"
     2 - 25 moves/hour"''
    " 3 - 30 moves/hour"

270 PRINT TAB(1,18)"4 -
     All moves in"'''"
        15 minutes"'''"
     5 - All moves in"''
    "     30 minutes"''
    " 6 - All moves in"''
    "     60 minutes"
280 ON INSTR("0123456"
    ,GET$ ) GOTO 360
    ,290 ,300 ,310 ,320
    ,330 ,340
    ELSE 280
290 Rate%=20
    :GOTO 360
300 Rate%=25
    :GOTO 360
310 Rate%=30
    :GOTO 360
320 All%=15
    :GOTO 350
330 All%=30
    :GOTO 350
340 All%=60
350 CLS
    :COLOUR 1
    :PRINT TAB(5,28)"Time
     limit:"'''"ALL MOVES/";
    All%;" minutes"
    :GOTO 370
360 CLS
    :IF Rate%<>0 COLOUR 1
    :PRINT TAB(1,28)"Move
     rate: ";Rate%;"/hour"
370 PROCdraw(600)
    :PROCdraw(1232)
380 COLOUR 3
    :PRINT TAB(3,4)"WHITE"
    TAB(2,8)"0: 0: 0"
    TAB(4,16)0
390 COLOUR 0
    :PRINT TAB(13,4)"BLACK"
    TAB(12,8)"0: 0: 0"
    TAB(14,16)0
400 COLOUR 1
    :PRINT TAB(8,14)"Moves"
410 PROCwait
420 SOUND 1,-10,93,5
430 REPEAT
440 PROCtime(0,1,8,3)
450 PROCtime(1,11,8,0)
460 UNTIL FALSE
470 ON ERROR OFF
480 IF ERR =17 RUN
490 MODE 7
    :REPORT
    :PRINT " at line ";ERL
500 @%=10
510 *FX4
520 *FX12
530 END
550 DEF PROCdraw(x%)
560 GCOL 0,1
570 MOVE x%,800
    :DRAW x%,700
580 DRAW x%-548,700
    :DRAW x%-548,800
590 DRAW x%,800
600 ENDPROC
620 DEF PROCwait
630 COLOUR 3
    :PRINT TAB(3,21)"Press
     SPACE BAR"'"   to
     start clock"
640 REPEAT UNTIL GET$ =
    " "
650 PRINT TAB(0,21)
    SPC (60)
660 ENDPROC
680 DEF PROCtime(N%,X%
    ,Y%,P%)
690 TIME =T%(N%)
700 REPEAT
710 X$=INKEY$ (0)
720 Z%=TIME
730 C%(N%)=Z% MOD 100
740 S%(N%)=(Z% DIV 100)
    MOD 60
750 M%(N%)=(Z% DIV 6000)
    MOD 60
760 H%(N%)=(Z% DIV 360000)
    MOD 12
770 COLOUR P%
    :PRINT TAB(X%,Y%)H%(N%)
    ":"M%(N%)":"S%(N%)
780 IF All%=0 GOTO 800
790 IF All%=M%(N%)
    OR All%=H%(N%)*60
     PROCbuzz
    ELSE 810
800 IF Move%(N%)<Rate%*H%(N
    %) PROCbuzz
810 IF X$="S" OR X$="s"
     PROCwait
    :TIME =FNtime(C%(N%)
    ,S%(N%),M%(N%),H%(N%))
820 UNTIL X$=" "
830 SOUND 1,-10,77+N%*16
    ,5
840 T%(N%)=FNtime(C%(N%)
    ,S%(N%),M%(N%),H%(N%))
850 Move%(N%)=Move%(N%)+1
    :PRINT TAB(X%+3,Y%+8)Mo
    ve%(N%)
860 ENDPROC
880 DEF FNtime(c%,s%,m%
    ,h%)
890 =c%+(s%*100)+(m%*6000)+
    (h%*360000)
910 DEF PROCbuzz
920 FOR I%=1TO 2
    :SOUND 1,1,97,6
    :SOUND 1,1,77,10
    :NEXT I%
930 IF N%=0 P$="WHITE"

    ELSE P$="BLACK"
940 PRINT TAB(5,21)P$+
    " LOSES"''"   ON TIME
     DEFAULT"
950 REPEAT UNTIL FALSE
960 ENDPROC
980 DEF PROCbuzz(Interval%)
990 REPEAT
1000 T%=TIME
1010 REPEAT UNTIL TIME =T%+I
    nterval%*100
1020 SOUND 1,-12,33,40
1030 UNTIL FALSE
1040 ENDPROC
```

This listing is included in this month's cassette tape offer. See order form on Page 43.

BRAINTEASER

About which book did *ELECTRON USER* say:

"each program an interesting and amusing challenge"

"a nice change - an enjoyable book, far from the run of the mill"

"if you like puzzles and you've got an Electron, then you'll like this book"

ANSWER BRAINTEASERS

This unique computer book, designed for the 15 plus age group will test your logic, general knowledge mathematical skills

Available from all good book shops or direct at £5.95 plus 55p p&p.

PHOENIX
PUBLISHING ASSOCIATES
14 VERNON ROAD BUSHEY
HERTFORDSHIRE WD2 2JL
TEL WATFORD 32109

NAME .....

ADDRESS .....

.....

..... POSTCODE .....

Cheques/Postal Orders to:- Phoenix Publishing Associates 14 Vernon Road, Bushey, Herts.

GARLAND COMPUTING

35 DEAN HILL - PLYMOUTH - PL9 9AF. TELEPHONE: 0752 41287

LEARNING MATHS with the electron

A collection of programs for use by children of 9 upwards. Written by a teacher and approved and used in schools throughout the country.
Each package contains 3 to 4 programs using animations and simple games to help learn the principles of maths in an interesting and entertaining way. All members of the family will enjoy using these programs.

A series of nine titles is available:-

JM1 Angles
JM2 Directed numbers
JM3 Fractions
JM4 Co-ordinates and lines
JM5 Symmetry
JM6 Motion geometry
JM7 Sets
JM8 Elementary statistics
JM9 Ratio

Each package is superb value at just £7.00 (inc. VAT and P&P)

Available by mail order, or from selected computer stores and educational suppliers. Please send for full details.

the educational specialists

SS

ELECTRON UTILITIES **£8**
15 useful programs and procedures that can save hours of programming effort. Includes statistical diagrams, super-fast sorting (much faster than bubble), colour graphics and lots more.

INFORMATION HANDLING **£11.50**
A two-cassette package of programs and data-bases to introduce you to the world of information technology. Use large data-bases, create and use cassette files, produce your own electronic dictionary. Features 19th century population survey with full documentation on how to computerise similar information for your own area. This package has been written for new computer users or anyone who wants an introduction to information handling.

FUN MATHS 1 and 2 **Age 6-13**
Two packages that have been designed by teachers and written by professionals to greatly improve the mental number work of their pupils. These full-graphics programs are great fun to use and have proved very successful in speeding-up logical thought and mental arithmetic.

1. SQUARE PUZZLES (4 programs) **£6**
2. INVADER MATHS (2 programs) **£4**

(Buy both for £8.50)

PRINT-AND-PLOT PAD
If you are going to write useful programs for the ELECTRON then careful planning of screen layouts is essential. Our 60-sheet pad allows rapid designing of displays, in both text and graphics, in all modes. Each sheet also includes user-defined-graphics grids.

GRAPH PADS **£2 each or 3 for £5**

Cheques etc. to:

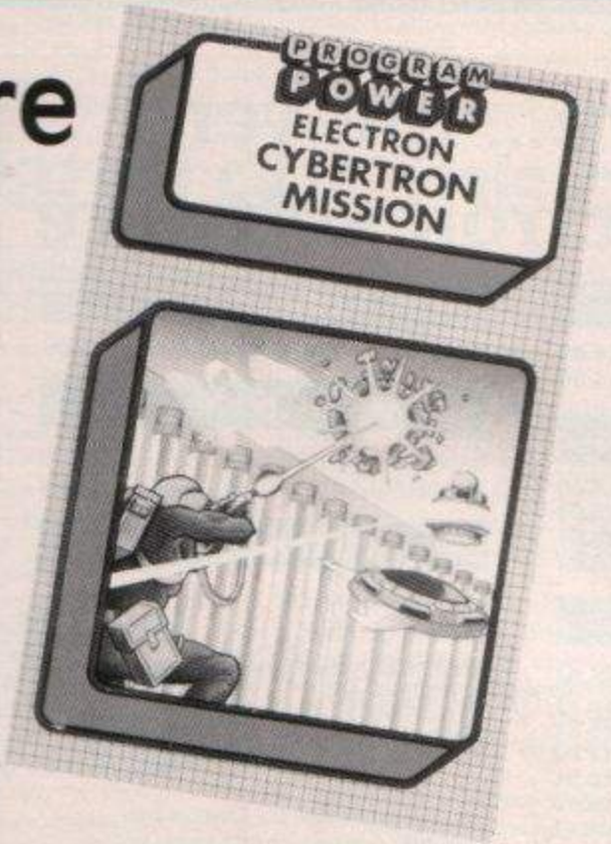
SCORBY SOFTWARE,
Main St., Flixton, Scarborough, YO11 3UB

Software Surgery

THE COLUMN THAT TAKES A LOOK INSIDE THE LATEST RELEASES

Adventure into an arcade winner

Cyberton Mission
Program Power



"ABSORBING", "Electrifying", "Frustrating", "Addictive" are just a few of the adjectives I would use to describe Program Power's latest space game.

Load the program into your Electron and you are immediately conveyed into a danger-strewn world of spinners, clones, cyberdroids and spooks. You may not be sure exactly what they are but you can be certain they're nasty.

On the first level you are instructed to find a key which can be used to open a safe.

Doing this conveys you to higher levels, where more dangers await you.

However things are not as straightforward as just wandering round the screen until you find the key. Life in space – or at least in space games – is never that simple.

You'll need every one of your five lives as you battle your way through a series of maze-like rooms.

You score points each time you zap a spinner, and gain an extra life when, and if, you reach a pot of gold.

If you survive the first few batches of spinners you'll find that clones begin to block your way.

And after the clones come the cyberdroids – vacuum cleaner lookalikes with nasty dispositions.

Two points to note. First, watch out for the spooks. These little treasures will come and get you at every possible opportunity. You've got to be alert and quick on the draw.

Second, you must have the key in order to open the safe to proceed to the higher levels.

The program is a cross between arcade-style action and an elementary adventure, combining the two perfectly.

It's exciting with plenty of variety, excellent graphics and interesting sound effects. You'll be a-mazed. And if you are anything like me, you won't be able to put your Electron down. A winner.

Paul West

Monster mission

Castle Frankenstein
Epic Software

CASTLE Frankenstein is a text adventure originally written for the BBC Micro B and has now been converted to run on the Electron.

The plot centres around Frankenstein's monster.

Originally he was thought to have perished in a fire 20 years ago. But now, because of unsolved murders in the area, there's a growing suspicion that he's alive and well and intent on vengeance.

The villagers have elected you to be their champion, and your task is to find and destroy the monster.

To help you the cassette comes with an insert which gives general information about the game.

Something I found rather strange about this insert was a claim that the tape was disc compatible. I suspect this was intended for the BBC rather than the Electron.

However it probably won't be long until the Electron has discs, so curiosity made me try putting the tape onto a friend's BBC Micro with discs.

I found you could not use it on disc without using a routine to move it down in memory.

Even then the save-game option would only work with cassette. I would be interested to know if the same is true of the BBC version.

On loading the program presents instructions and background information. Then begins one of the best all-round adventures I have ever seen for the Electron.

I will not reveal anything about the actual playing of the game. That's a pleasure I'll let you experience for yourself.

Whoever wrote this program has an extremely devious mind, and makes you work very hard for each piece of progress.

Yet, at the same time, he allows you to roam quite a distance before presenting you with puzzles to solve. This, I feel, is the proper way to write an adventure.

The beginner has lots of locations to explore to get the feel of the game, but the more experienced adventurer can go through them rapidly to reach the puzzles.

There were a few minor things I wasn't happy with. For instance, there's no on-screen indication of exits. But I've probably just got into lazy habits with other adventures.

The program itself responds very quickly to keyboard input, and the save-game facility – which is an absolute necessity – is very fast.

Overall, an extremely good adventure and excellent value for money.

Merlin

The frogs march on

Croaker
Program Power

CROAKER is another version of that well known game in which suicidal frogs cross busy highways and then hop their way to safety across a river in order to reach a hole in the bank.

One day I am going to ask someone how come frogs drown if they fall into a river?

The program loads reliably and screen instructions appear while the main code is being loaded in.

The configuration of the

→

National Micro Centres

One-stop shopping for all that's best for the Electron

Telephone orders: 061-429 8080 (24 hours)

It's all here... the very latest Electron software

LANGUAGES

Forth (AS)	£14.00
Lisp ' (AS)	£14.00

EDUCATION

Draw (MP)	£8.65
Invisible Man (CS)	£6.04
Puncman 1 and 2 (CS)	£8.04
Tree of Knowledge (AS)	£8.00

RECREATION

Bandits at 3 O'Clock (MP)	£6.04
Chess (MP)	£6.91
Chuckie Egg (AF)	£6.87
Crime & Detection Quiz (AS)	£11.00
Croaker (MP)	£6.91
Cybertron Mission (MP)	£6.91
Cylon Attack (AF)	£6.87
Draughts & Reversi (AS)	£8.00
Escape from Moonbase Alpha (MP)	£6.91
Felix & The Fruit Monster (MP)	£6.91
Felix in the Factory (MP)	£6.91
History Quiz (AS)	£11.00
I Do (AS)	£11.00
Intergalactic Trader (MP)	£7.78
Kamakazi (AF)	£6.87
Killer Gorilla (MP)	£6.91
Meteors (AS)	£8.00
Monster (AS)	£8.00
Moonraider (MP)	£6.91
Music Quiz (AS)	£11.00
Pharaoh's Tomb (AF)	£6.87
Positron (MP)	£6.04
Royal Quiz (AS)	£11.00
Science Fiction Quiz (AS)	£11.00
Starship Command (AS)	£8.00
Swoop (MP)	£6.91
The Dating Game (AS)	£11.00
What Makes You Tick (TP)	£5.17

AF – A&F Software
AS – Acornsoft
CS – Chalksoft
MP – Micro Power
TP – Third Program

Personal shoppers can obtain the products advertised on this page from the following retail stores:

Stockport Micro Centre
4/6 Brown St., Stockport,
Greater Manchester.
Tel: 061-480 0539

Wilmslow Micro Centre,
62 Grove Street,
Wilmslow, Cheshire.
Tel: 0625 530891

BUSINESS & HOME

Money Management (AS)	£10.00

GRAPHICS

Creative Graphics (AS)	£8.00
Graphs and Charts (AS)	£8.00

DATA STORAGE

A data cassette recorder that perfectly matches the Electron

£34.74

Fully compatible with the Electron and specifically designed for data saving and loading, the Pye data cassette recorder (with free power pack and Electron lead) is guaranteed to end your data storage and retrieval problems.

UK CUSTOMERS: Add 15% VAT. carriage free. Overseas Sales VAT free. please enquire for carriage rates.

Be among the first to own the electrifying Electron!

The baby brother of the BBC Micro has been hailed by the computer press as a big breakthrough in power and price. This exceptional machine is already being forecast to be the top-selling micro of 1984. It comes complete with an introductory cassette of 15 programs, a very comprehensive user guide and an easy to understand DIY book on programming.

£173.00

Stocks are limited, so it's very much a case of first come first served. We promise that no cheques will be cashed until the machine is despatched.

Electron Dust Cover **£2.56**

ORDER FORM

Post to:
NATIONAL MICRO CENTRES,
36-38 St. Petersgate,
Stockport SK1 1HL.

Please supply the following:

Item	Qty	Total £	p

VAT 15% .....
Carriage free .....
TOTAL

Attractive credit terms
Phone for details

Please indicate method of payment:

☐ Cheque payable to National Micro Centres

☐ Access/Barclaycard No.

☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Name .....
Address .....
.....
.....
Tel. No. .....
Signed .....

EU

From Page 27

keys is a little unusual – A and Z for up and down, while M and N control lateral movement.

However, they soon feel natural enough, although I would imagine that a joystick would improve matters.

The game's format is fairly standard, with five lanes of traffic travelling in alternate directions and at different speeds.

After a brief rest on the riverbank, there are then five more lanes of logs and turtles before safety is reached at one of the five holes.

When all are occupied bonus points are gained and the screens become more difficult. The cars move more quickly and are more frequent.

Things are even worse in the river. Some of the turtles dive, and many logs turn out to be crocodiles with gaping jaws.

The game has little to make it stand out from its clones. But the graphics are quite presentable, with good use of colour. The key response is quick and positive.

I appreciated the first screen starting at a very easy level – my six-year-old son was able to do well at this initial level, although the crocodiles made him ditch many frogs into a watery grave.

Too often, a game starts with a level of difficulty that doesn't allow the young or inexperienced to achieve any success.

Here it is possible to gain practice on the lower levels to help mount an attack on the author's claimed top score of 12,530.

This is a competent and addictive version, but without special features.

Probably the most used facility will be that which turns off the awful tune and reverts to the original sound effects.

Phil Tayler

Electron User index of software reviews

Watch out, this caterpillar is carnivorous

Caterpillar
IJK Software

HAVE you ever felt the need to destroy a defenceless caterpillar? If you haven't so far, now's your chance.

In a variation of the popular arcade game, you control the black, movable weapon at the bottom end of a field of mushrooms.

You are hungry for points. The caterpillar is at the other end, hungry for you.

Hang on to your nerve as you watch it menacingly winding its way towards you, weaving between the mushrooms.

As you move from left to right or up and down you fire at the caterpillar, blasting mushrooms out of the way, scoring points all the time.

When you hit the lengthy beasty, a segment is destroyed. If you hit it in the centre then it splits into two.

But it still comes towards you. Can you destroy it before it gets you?

While you're watching it come closer, you mustn't forget to fire at a scorpion which occasionally appears. A lot of points can be gained from hitting that particular undesirable.

Watch out, too, for a spider. He's hanging around the bottom of the screen and ready to grab you if you can't shoot or avoid him.

A nice little game, one that has everyone in the room wanting a go – while you're reluctant to let them. Graphics and sound effects are well up to standard.

Graham Parr

With the accent on action...

IF you're one of these shady characters who can go in a pub or amusement arcade and lose yourself for hours in a Space Invaders or Galaxians game, then this should be right up your street.

There are fast and slow levels – and you take your pick according to how big-headed you feel. Then launch into the fray.

You are a lone, ground-based, tank-like vehicle fighting squadron after squadron of aircraft, all intent on sending you to the big electron cloud in the sky.

The skill lies in dodging the bombs and the descending bombers, who have no fear of ramming you.

Kamikazi
A&F Software

At the same time you are trying to shoot them down. But to add insult to intended injury once you've annihilated one squadron another more challenging one is ready to take its place.

This is not the most original game in the world, but it is certainly well done.

The action is fast and furious with more than adequate sound and graphics.

If you are looking for a classic game to test your nerve and reactions this is for you.

Peter Gray

Are you fed up with shooting aliens, jumping barrels, or hopping over rivers? You are? Then try an adventure

AN adventure is a fantasy world which you, the hero, have to explore, usually with the object of finding treasure or rescuing princesses, and generally being a hero.

Kids stuff? Not at all.

The crafty programmer who's written the game doesn't want you to win too easily. So he makes it as hard as possible, which is often very hard indeed.

Believe me, when you've spent an hour trying to find a key to open a mysterious locked door only to find that the door is locked from the other side, you'll be ready to strangle that programmer.

An adventure is like a detective novel, full of clues, puzzles and red herrings. Your job is to sift the clues, solve the puzzles and, hopefully, recognise the red herrings.

What's more, because you're in a fantasy world, with its own natural laws, you can also have goblins, magic or even aliens to cope with.

Not quite that easy after all, is it?

So where do these adventure games come from? They owe their origins to the Dungeons and Dragons craze that swept America in the mid-1970s.

Two mainframe programmers, Crowther and Woods, wrote a program called Colossal Cave, which simulated a D&D game, but had more emphasis on problem solving and less on fighting monsters.

This quickly achieved cult status among other programmers, and might have remained on mainframes but for an enterprising man called Scott Adams.

He adapted one of these massive programs to a 16k TRS-80, published it, and the first adventure for a home micro, Adventureland, was released.

Since then many adventures have been written. They can be split into two basic types – graphic and text.

Graphic adventures get their name more from the graphic action in them than the pictures on the screen, though they generally do have graphics of some kind.

They tend to simulate a D&D game very closely, in that you choose the type of role you wish to play, such as warrior, cleric, barbarian, wizard and so on.

On the basis of your choice you're assigned strengths and weaknesses which you exploit to achieve the objectives set in the adventure, like collecting treasure.

Since this treasure is almost invariably in the possession of some monster or other you spend most of your time fighting them. The result is that your progress often seems to depend more on luck than skill.

Text adventures earn their name because they originally consisted of text only, and were based on the same type of format as the original Crowther and Woods game.

Obviously there are now adventures with both text and graphics, so we can say that a strong sword arm is necessary for a graphics game and a lot of thought for a text game.

In this article I shall only be dealing with text adventures.

If they have their own history and are considered to be so good how come you haven't heard of them?

We all know about arcade games, and there are some brilliant versions available for the Electron.

But there are no adventure games in the arcades, so you either come across them by chance or somebody recommends them to you.

You either love them or hate them, and it's very hard to drag away the adventure fanatic from his machine long enough to talk about them.

You must have seen one of these adventure freaks. They're the ones who come to the computer club bleary-eyed from playing their latest game until three in the morning.

Yes, I know you thought he was an insomniac, but now you know.

What's so special about these adventure games?

I gave you an idea earlier of the object of them, so let's give you an example from that first Scott Adams game.

The aim is to collect and store 13 treasures. To get one of them you have to wake a sleeping dragon with some bees.

The bees have to be caught in an empty bottle – after you have first covered yourself in mud to stop them stinging you.

The bottle is full at first and has to be emptied over some lava to get another treasure.

However once you get to

BE ADVENTUROUS WITH YOUR ELECTRON

the location where you empty the bottle you need a rug and a magic word to get out. To get the rug you need to rub the lamp in another location.

Not only that, you have to climb down a hole to get the means to light the lamp, which you find by chopping down a tree, after you've first climbed it to get the key which opens the door . . .

Phew! Bit involved isn't?

But that's where the attraction lies, in solving the puzzles, progressing through the locations and getting that final message on the screen: "CONGRATULATIONS! YOU ARE A MASTER ADVENTURER!"

I know it must seem very complicated, but adventures are totally logical. Admittedly that logic is sometimes very obscure but all the puzzles can be solved.

And there is no greater feeling than to solve a problem that has been stumping you for hours.

Now I've got you interested in them and you're all going to rush out and buy up the shop, let me give you the bad news: ALL adventures are very hard for ALL beginners.

The good news is that they are just like everything else. The more you do them, the better you get. I well remember my first game, and I can assure you it was not a very auspicious beginning.

However there are certain things common to most adventures. I will explain how to cope with them so that your first game won't be quite as traumatic as mine.

Most, if not all, adventures have a maze in them somewhere. Often these mazes are logical, so if you go North and then South you end up in the location you started from.

Others are not so logical, but the answer for both is the same – make a map.

If you cannot recognise your location from the objects present, room description or the direction of the exits, then drop some of your own objects and make a map based on them.

Some adventures have more than 200 locations, so it is a good idea to make a map of your travels anyway.

Another thing common to most adventures is ending up in the dark, often underground or in unlit rooms. Obviously you need to get a lamp or torch or at least some matches.

Should you come across one in your travels always check to see if you can light it first. Do you need matches or batteries – or oil if it's an oil lamp?

If you do end up in PITCH DARKNESS, try and reverse the move you have just made. If that proves fatal, try and find the lamp and the means of lighting it before you re-visit that location.

If you've got the lamp, try LIGHT LAMP or ON or anything else you can think of before moving.

A few other things that might help you which should be obvious are to do with shovels, scenery and ropes.

If you find a shovel it's a good bet that you will have to DIG somewhere, either to find a treasure or to get an object that will help you somewhere else in the adventure.

Examine your surroundings. If you are in a forest, can you climb a tree? Or if you've got an axe, can you chop that tree down? Can you climb a wall, or a statue?

If you find a rope it's likely to be needed somewhere, either to climb something or perhaps to pull something.

An object that is too heavy to lift might be pulled if you TIE ROPE and PULL the object.

There are some general tips that are applicable to all adventures.

If the program allows you to save the game – that is, allows you to return to the location you have reached should something you do prove fatal – then use it before you enter any suspicious places, or before trying something dangerous.

If something doesn't work, such as taking a bucket stuck in the mud by keying in TAKE BUCKET, then try doing it a couple of times.

These programmers are a crafty bunch, and sometimes make you do a thing a few times before you succeed.

Always read the room descriptions very carefully, sometimes clues are hidden here. Always EXAMINE everything.

I hope you now have an idea of what adventuring is all about. You never know, maybe we'll be seeing *you* staggering into the computer club with bleary eyes sometime.

Happy adventuring!

MERLIN

SOUNDS.. EXCITING

BUILD up a library of exciting sounds to enhance your own programs with these listings. And many more in the months to come!

PAYTONE

From I.G. Fothergill

```
10 REM PAYTONE
20 FOR A=1 TO 20
30 SOUND 1,-15,129,2
40  SOUND 1,0,129,2
50 NEXT
```

ENGAGED TONE

From I.G. Fothergill

```
10 REM NUMBER
20 REM BUSY ONE
30 FOR B=1 TO 15
40 SOUND 2,-15,129,7
50 SOUND 2,0,129,7
60 NEXT
```

EXCHANGE BUSY TONE

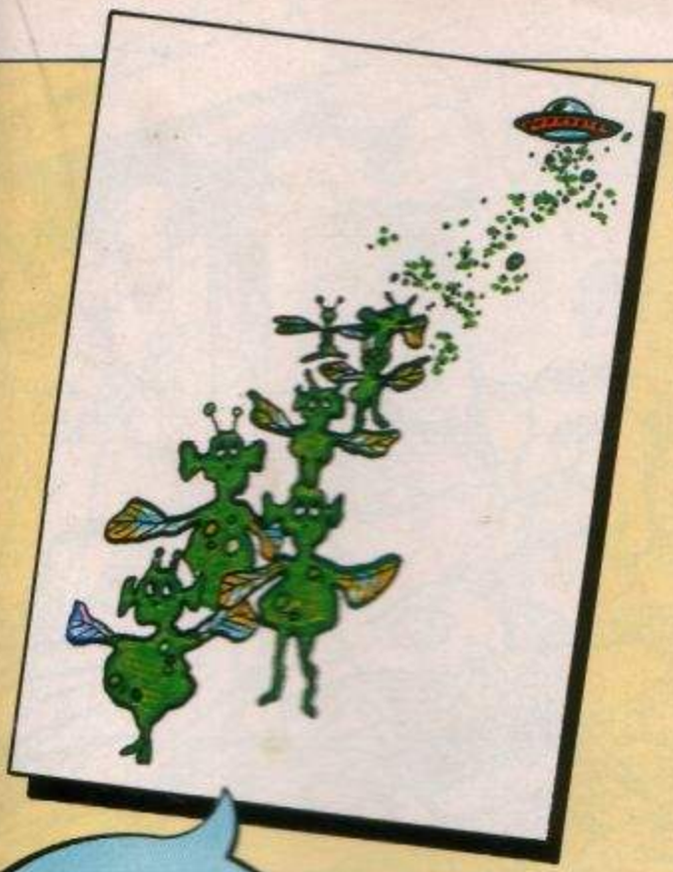
From I.G. Fothergill

```
10 REM BUSY TONE
20 FOR C= 1 TO 10
30 SOUND &01,-15,149,4
40 SOUND &01,0,149,10
50 SOUND &01,0,149,10
60 SOUND &01,0,129,7
70 NEXT
```

CUT OFF

From Anthony and Melenie, Bradford

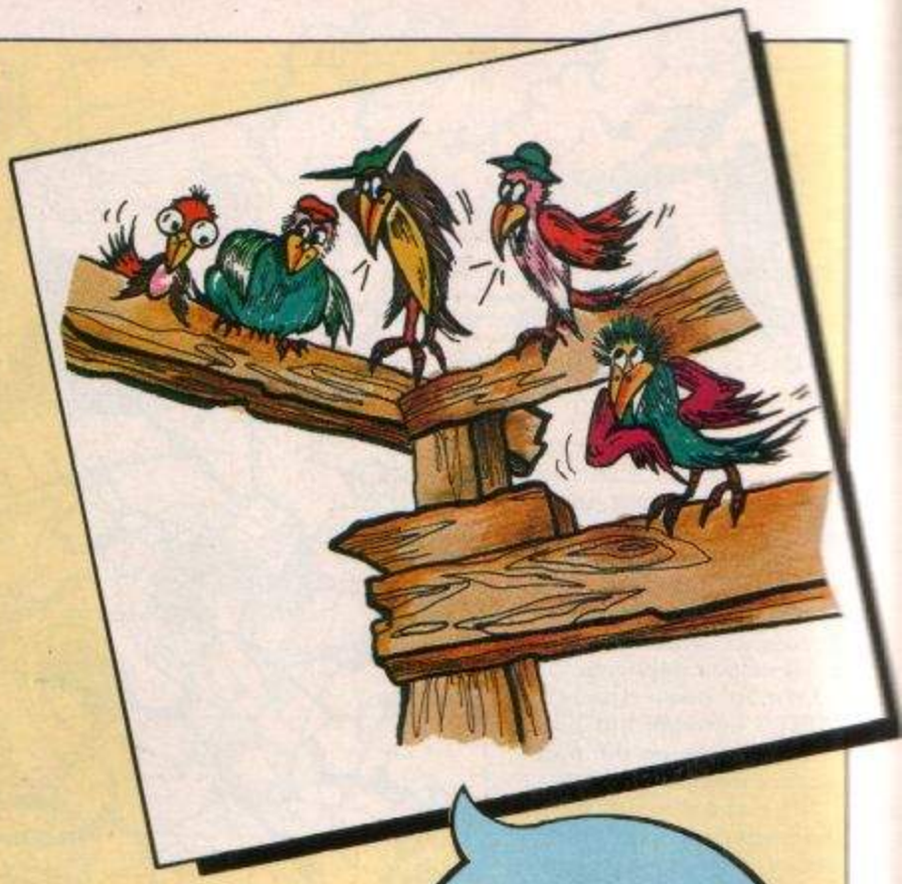
```
SOUND 1,-15,129,1000
```



ALIEN SWARM

From John Ward

```
ENVELOPE 1,1,127,126,0,81,
47,0,0,126,0,0,-126,126,126

SOUND 1,1,155,53
```



CRAZY BIRDS

Anonymous

```
ENVELOPE 1,7,77,7,77,7,77,7,
126,0,0,-126,126,126

SOUND 1,1,200,200
```

CAR ALARM

Anonymous

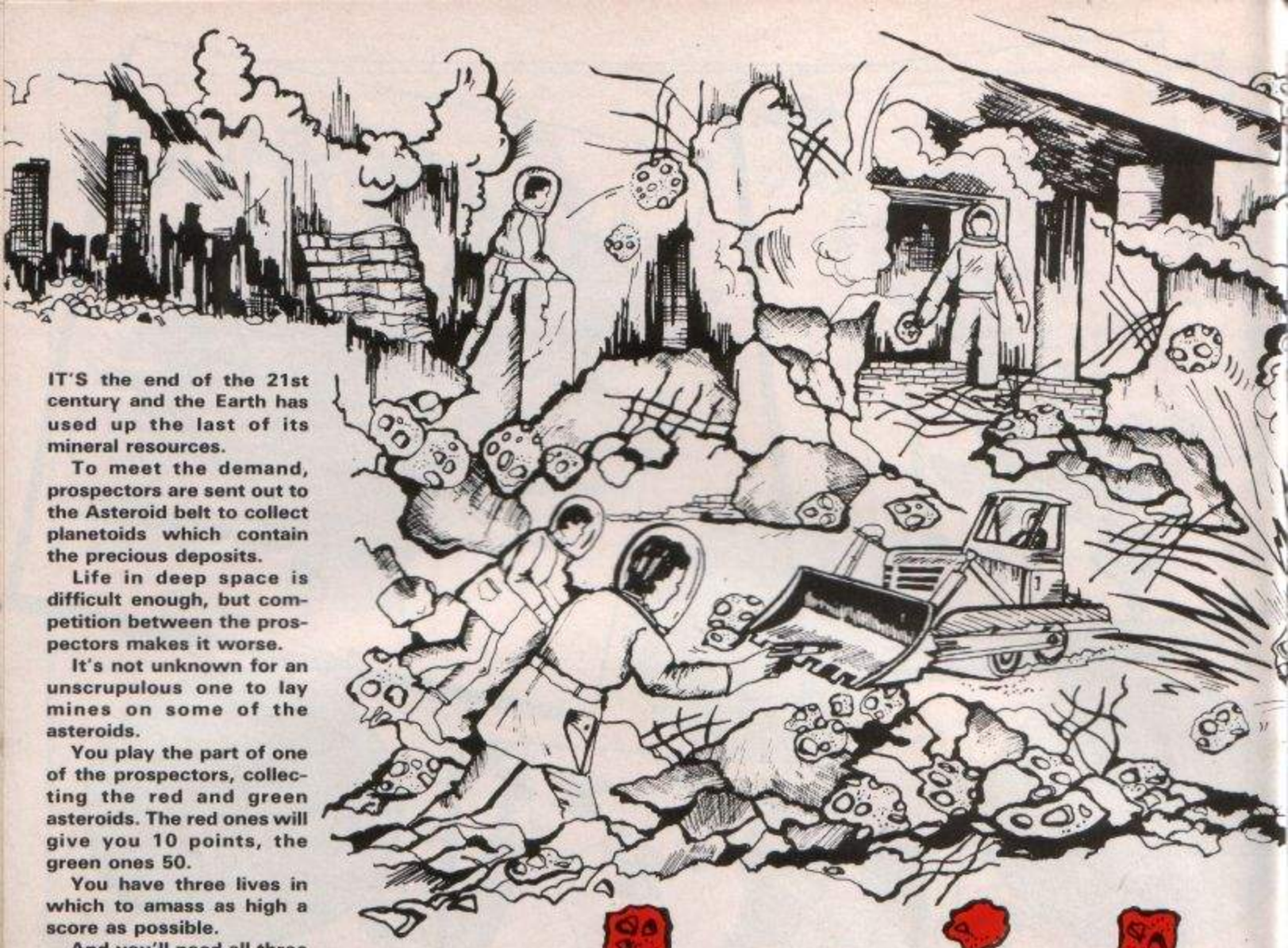
```
ENVELOPE
1,2,3,4,5,6,7,8,126,
0,0,-126,126,126

SOUND 1,2,200,200
```

PINBALL

From Peter Dobbs

```
ENVELOPE 4,5,-45,1,2,-9,0,0,
126,0,0,-126,126,126

SOUND 1,4,26,26
```

Do you have any sounds for Sounds Exciting? Send them into Electron User and hear yourself in print. The address: Sounds Exciting, Electron User, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.



asteroids

By ERIC H. CRISP

IT'S the end of the 21st century and the Earth has used up the last of its mineral resources.

To meet the demand, prospectors are sent out to the Asteroid belt to collect planetoids which contain the precious deposits.

Life in deep space is difficult enough, but competition between the prospectors makes it worse.

It's not unknown for an unscrupulous one to lay mines on some of the asteroids.

You play the part of one of the prospectors, collecting the red and green asteroids. The red ones will give you 10 points, the green ones 50.

You have three lives in which to amass as high a score as possible.

And you'll need all three because if you collide with an asteroid, or pick up a mine and can't defuse it, you explode into a cloud of interstellar dust.

It's up to you. Remember, the Earth needs those minerals.

Good luck!

PROCEDURES

Booby	Tests for correct combination entered.
Crash	Ship explosion.
Hit	Tests whether an asteroid has crashed into the ship or been collected. The score adjusted.
Initial	Initialises characters and dimensions arrays.
Instruct	Displays the instructions.
Moveast	Moves the asteroids.
Play	The main procedure.
Plot	Plots an asteroid.
Rocket	Displays the rocket exhaust.
Rotate	Calculates new positions and directions due to rotation.
Setup	Sets up the screen display for each new ship.

CHARACTER DEFINITIONS

Asteroids	Character 240.
Spaceship	Characters 241 to 248.
Rocket exhaust	Character 249.
Lives	Character 250.

VARIABLES

A%	Score.
A$	Combination input.
C%	Plotting colour.
D%	Level of difficulty. As game progresses it increases the possible speed of the asteroids.
DX%(I%)	Horizontal speed of an asteroid.
DY%(I%)	Vertical speed of an asteroid.
F%	Finished flag.
G%	Lives.
I%	Asteroid counter.
J%	Loop counter.
K,L	Multiplication factor for rotation calculations.
KEY$	The three letter combination.
R%	Rotation direction of ship –1, 0 or 1.
S%	Ship's speed – 0 or 1.
T%(I%)	Asteroid type. 1 = red, 2 = green, 6 = mine.
X%(I%)	X coordinate of an asteroid.
Y%(I%)	Y coordinate of an asteroid.
Z%	Time for entering the three letter combination.

PROGRAM CHANGES

The game can be made harder or easier by making some or all of the following changes:

Time allowed for defusing a mine: The total time allowed for this is given by the value of Z% in line 130.

The time allowed for inputing one letter is governed by the INKEY value in line 150. Increasing these values allows more time.

Asteroid speed: This is governed by the value of D%. The larger its value, the faster the asteroids move.

It is initialised on line 40 and is steadily increased through the game on line 330.

Asteroid collection: The first comparison on line 300 affects the accuracy with which asteroids must be collected.

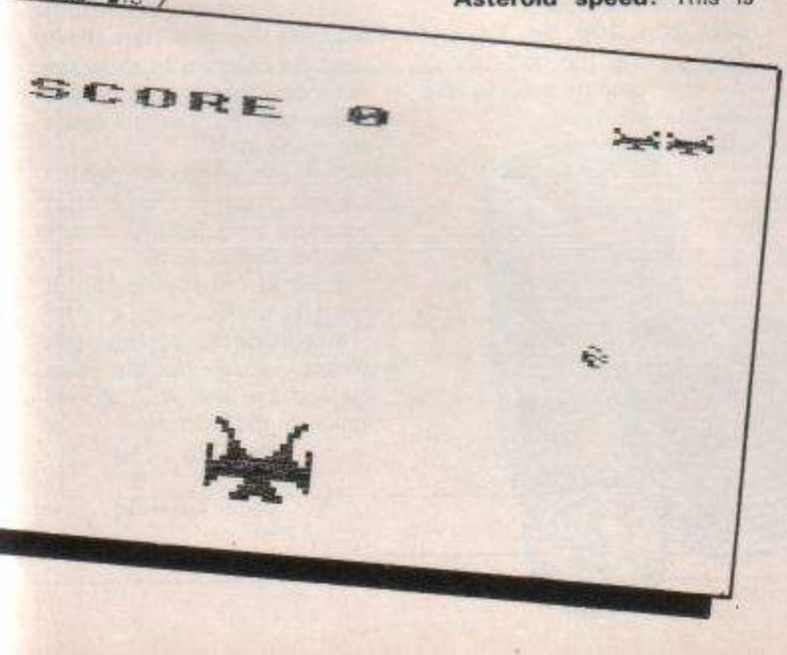
Collection is made more difficult by decreasing the number that is compared with ABS(X%(I%)+16).

Number of asteroids: The maximum number that can appear at one time is governed by the range of I%. Its range is given by the FOR NEXT loops that start on lines 670 and 930.

Note that the program has been written for five asteroids. If more are desired then the DIM statement on line 390 must be changed to allow larger arrays – that is, the numbers in the brackets must be changed.

Also note that more asteroids will slow the game down due to the extra processing time needed.

Lives: The number of lives is held in G% which is initialised on line 40.




```
10 REM ASTEROIDS
20 REM (C) ELECTRON USER
30 MODE 6
   :PROCInstruct
   :PROCInitial
40 REPEAT
   :G%=3
   :A%=0
   :D%=90
50 REPEAT
   :MODE 5
   :PROCSetup
   :PROCPlay
   :UNTIL F%
60 MODE 6
   :PRINT TAB(5,10)"You
    scored ";A%
   :*FX15,0
70 PRINT TAB(5,15)"PRESS
    A KEY FOR ANOTHER GAME"'
   '"     OR RETURN FOR
    INSTRUCTIONS"
80 IF GET =13
   THEN RUN
90 UNTIL 0
100 REM **** Booby ****
110 DEF PROCBooby
120 KEY$=CHR$ (RND(26)+64)+
    CHR$ (RND(26)+64)+
    CHR$ (RND(26)+64)
    :VDU 4
130 TIME =-200
    :Z%=100+RND(400)
140 SOUND 1,1,100,255
    :A$=""
    :REPEAT UNTIL TIME >0
    :PRINT TAB(0,0)KEY$
    :SOUND 17,1,150,255
    :*FX15,1
150 REPEAT
    :A$=A$+INKEY$ (200)
    :UNTIL A$=KEY$OR
    TIME >Z%
    :*FX15,0
160 IF A$<>KEY$
    THEN PROCCrash
```

Full listing starts on Page 57

This maths workout is based on articles that originally appeared in The Micro User. Our thanks to our "big brother" magazine for permission to use it.

MIKE BIBBY'S

MATHS workout

Exercises for the Electron

WELCOME to the first in a series of articles in which we hope to take the mystery out of understanding the fundamentals of the Electron's workings.

All too often even competent Basic programmers tend to shy off such topics as binary coding, hexadecimal and assembly language because it seems too "mathematical".

This is a great pity, because the Electron is so constructed that a little knowledge in these fields allows you to take full advantage of its advanced facilities.

The mathematical aspects of the subject aren't at all deep. Certainly anyone who can follow Basic should be able to cope with this series.

If you feel that despite our best efforts we still haven't explained something fully enough, please write in and tell us. We'll try to rectify the situation in later articles.

First we are going to look at binary code.

This is a way of handling numbers essential to our understanding of what goes on inside a computer.

Binary is just a way of coding numbers in a way particularly suitable for computers. It's actually quite simple.

What often confuses beginners is the fact that the binary system codes numbers in a way that can look extremely like the way we normally code numbers.

For example, if you were presented with a number 100, you would probably decode it in your normal way and say it was "one hundred".

That, however, is just one way of interpreting it. If you decided to decode it as a binary number, you would interpret 100 in a completely different way and say it meant the number "four".

(Never mind exactly how you arrived at that conclusion for the moment.)

This is what often causes problems. People are so used to dealing with their numbers in the normal way that 100 is always "one hundred" to them. They can't make the shift necessary to decode it in binary as "four".

Actually it is rather ambiguous. Presented with 100, do you interpret it as "one hundred" or "four"?

Our rule will be, if you mean our usual way of dealing with numbers *(the hundreds, tens and units you learnt at school – or to put it more formally, the denary system)* you write the number in the normal way.

If you wish the number to be decoded as a binary number you put the symbol % in front of it. So 100 means "one hundred" while %100 means "four".

So far so good. We now have a marker (%) to warn us that we have to decode the number in a special way as a binary number.

However before you decode you need a rule for decoding. So how do you get the number "four" from %100? What's the rule?

Let's take a detour for the moment, and think about the coins we use every day. Our currency, until recently, consisted of these coins: **50p, 20p, 10p, 5p, 2p,** and **1p** (ignoring the half-pence). We can combine them to give any sum we wish.

For example:

75p is **50p** + **20p** + **5p** or **50p** + **10p** + **10p** + **5p** and so on.

We are all familiar with this. Often we use multiples of coins to make up a sum. For example, 5p can be **2p** + **2p** + **1p**.

Using the same coin twice, though, often means that we end up carrying unnecessary amounts of change. I for one don't like doing that.

Sometimes, however, with our present coinage system we have to use the same coin twice to obtain certain sums.

You cannot, for instance, make up the sum of 4p without doubling up coins. To avoid repeating coins we would have to invent a 4p coin.

Let's do that. In fact, let's invent a coinage system where you never have to use the same coin twice.

First of all we would need a **1p** coin and, of course, a **2p** coin. We cannot use **1p** + **1p** for 2p because it breaks the rule!

Now 3p can be made up of **1p** + **2p**. But for 4p we'll have to invent a **4p** coin.

Equipped with that we can make 5p (**4p** + **1p**), 6p (**4p** + **2p**), and 7p (**4p** + **2p** + **1p**).

In obtaining 7p we used all our available coins, so now we have to invent an **8p** coin.

If you work it out – and I suggest you have a go – you will find that with the coins you have at your disposal (**8p, 4p, 2p, 1p**) you can make any sum up to 15p. Then you would have to invent a new coin, **16p.**

Notice how the coins we have created have doubled in value: **1p, 2p, 4p, 8p, 16p.** No prizes for guessing what the next one is.

Let's summarise our results in a table (Figure I). Here I have used the columns to show the coins available and the rows to show how the various totals are made up.

A 1 in a particular column means that we use that column's coin, and 0 means that we don't use it.

Look at the row for 5p. It has 101 on it.

According to our rule this means we pick out the coins **4p** and **1p** (and NOT **2p**) to make up the 5p total:

	4p	2p	1p
%	1	0	1
→	4p + 1p = 5p		

TOTALS	COINS			
	8p	4p	2p	1p
1p				1
2p			1	0
3p			1	1
4p		1	0	0
5p		1	0	1
6p		1	1	0
7p		1	1	1
8p	1	0	0	0
9p	1	0	0	1
10p	1	0	1	0
11p	1	0	1	1
12p	1	1	0	0
13p	1	1	0	1
14p	1	1	1	0
15p	1	1	1	1

Figure I

Denary Value	Column or Bit Values 8	4	2	1	Binary Value
1				1	%1
2			1	0	%10
3			1	1	%11
4		1	0	0	%100
5		1	0	1	%101
6		1	1	0	%110
7		1	1	1	%111
8	1	0	0	0	%1000
9	1	0	0	1	%1001
10	1	0	1	0	%1010
11	1	0	1	1	%1011
12	1	1	0	0	%1100
13	1	1	0	1	%1101
14	1	1	1	0	%1110
15	1	1	1	1	%1111

Figure II

Now let's get back to computers by dropping all this talk about coins and redraw Figure I to show the same information but without referring to money – just numbers.

Figure II is the new table. As you can see, there is little change.

We can use this table to encode numbers in general, not just coins. We call this method of encoding the binary system.

Remember, to show that we mean a binary number we precede it with %.

So if you see, for example, %101 means:

```
    4  2  1
  % 1  0  1
→   4 + 1 = 5
```

That is, we add together the values of the columns containing 1. Look at row 5 of the table to check it.

Similarly, %1101 would mean 13 in the denary system since:

```
    8  4  2  1
  % 1  1  0  1
→   8 + 4 + 1 = 13
```

By now you should be able to work out for yourself why %100 represents four.

From the table, or by using the addition method I've just illustrated, see if you can decode the denary values of the following binary numbers:

%1001
% 101
% 11
%1101
% 111

You can use the program accompanying this article to check your results.

You've probably noticed by now that in the binary system you only use two symbols, 0 and 1, to encode numbers. Hence binary, bi–for two, as in bicycle.

You can encode any number that you want in binary. Just use more columns (or "bits" as we say in computer jargon), remembering that each new bit is worth double the preceding bit.

However it does get terribly cumbersome. For example, 100 (denary) encoded in binary is %1100100 since:

```
  64 32 16 8 4 2 1
% 1  1  0  0 1 0 0
→ 64+32+4=100
```

It is much easier to handle the number in our normal system.

To a computer this presents no problem. The fact that binary only uses two symbols is a bonus because you can represent numbers with a sequence of "switches".

Switches are what we call "two state". They're either ON or OFF.

If we have a sequence of four switches together we can encode numbers by having them either ON or OFF.

We could use ON to mean a 1, and OFF to mean a 0 in a particular column:

```
  8   4   2   1
ON OFF ON  ON
→%1011=11
```

Each of these "switches" represents a bit, and a computer memory is full of bits.

The 6502, which is the microprocessor at the heart of the Electron, deals with 524,288 of them.

To make things simpler, the 6502 handles the bits in groups of eight bits at a time – the group of eight being called a byte.

With this type of organisation the largest number you can store in a byte is 255 since:

```
  128 64 32 16 8 4 2 1
% 1   1  1  1  1 1 1 1
→ 128+64+32+16+8+
    4+2+1=255
```

Of course the computer can handle larger numbers *(and not just whole numbers)* but to do so it must use more than one byte.

Converting a byte from binary to denary is fairly straightforward. Simply write it down under the appropriate column (or bit) values and add together the value of all the columns in which a 1 occurs.

For example, given %10010101 you translate as follows:

```
  128 64 32 16 8 4 2 1
% 1   0  0  1  0 1 0 1
→ 128+16+4+1=149
```

Going from denary to binary is not at all difficult, but it is rather hard to put into words.

You do it by subtracting from the number you want to encode the value of each column in turn, starting with the highest (i.e. 128, 64, 32 and so on).

If you can subtract a particular column value you put a 1 in that column and continue to subtract the next lower column value from the remainder.

If you cannot manage the subtraction you put a 0 in that column and try to repeat the subtraction with the next lower column number.

So, starting with the highest column number (128 in our case), you:

REPEAT

1. Attempt to subtract the

→

149	
−128	128 goes – set to 1
21	64,32 can't go – set to 0
−16	16 goes – set it to 1
5	8 can't go – set to 0
4	4 goes – set to 1
1	2 can't go – set to 0
−1	1 goes – set to 1
0	

128	64	32	16	8	4	2	1
1							
	0	0					
			1				
				0			
					1		
						0	
							1
% 1	0	0	1	0	1	0	1

Figure III

From Page 37

relevant column number (highest first).

2. IF you succeed then put a 1 in that column number and continue to subtract other columns from the remainder. ELSE put a 0 in that column.

UNTIL all eight columns are covered.

Figure III should make it clearer.

In practice, when faced with encoding a number from denary to binary I tend to do it in my head, seeing which column values will add together to make the sum required, starting with the highest first.

For example, if I were to encode 161 in binary I would say, "Well, I can use 128, so that leaves me 33 to find. 33 can be made up of 32 and 1 so that does it: 128+32+1=161.

So I encode it as:

128 64 32 16 8 4 2 1
% 1 0 1 0 0 0 0 1
=%10100001

After a while you'll find this way quite simple.

To finish off, I'll leave you with a program to print out the binary value of a number between 0 and 255 (i.e. that can be stored in one byte).

Try it with various values and see if you can accept the results.

The program itself uses one or two ideas, such as AND, that may not be too familiar to you as yet.

Worry not. Watch these pages.

This listing was produced using a special formatter which breaks one program line over several lines of listing. When entering a line don't press Return until you come to the next line number. Full details of the formatter is given on Page 4 of the February issue.

```
10 REM ********************
   **
20 REM * ELECTRON USER
    '84 *
30 REM ********************
   **
40 MODE 6
50 ON ERROR GOTO 230
60 REPEAT
70 *FX15,1
80 CLS
90 @%=4
100 REPEAT
110 PRINT TAB(0,5)CHR$ (130)
120 PRINT TAB(1,5);
    STRING$(15," ")
130 INPUT TAB(1,5)"Denary
    "denary%
140 UNTIL denary%>=0
    AND denary%<256
150 PRINT TAB(1,12)"%"
160 FOR I%= 7 TO 0
    STEP -1
170 PRINT TAB(30-4*I%
    ,10)2^I%
180 PRINT TAB(30-4*I%
    ,12)(2^I% AND denary%)/
    2^I%
190 NEXT
200 PRINT TAB(0,20);
    CHR$ (132)CHR$ (157)
    CHR$ (131)"SPACE TO
    CONTINUE,ESCAPE TO
    END"
210 REPEAT UNTIL INKEY (-99)
220 UNTIL FALSE
230 END
```

This listing is included in this month's cassette tape offer. See order form on Page 43.

FRIEZE THAT SCREEN!

ALAN PLUME shows you how to create effective screen patterns by drawing just one figure

THIS program produces a frieze, a repeated pattern like the one pictured here. A frieze in its most basic form is simple to produce on an Electron, as it is merely the repeated drawing of one figure.

Creating the figure is probably the most difficult part to understand. Here it is made up of 25 user defined characters, listed in the DATA statements at the end of the program.

By altering them, you will be able to produce your own friezes.

As you'll discover, the black side borders are introduced to mask out the screen wrap-round which occurs when printing characters with the text and graphics cursors joined.

Why not use the program to produce your own friezes? You could make a fortune designing your own wallpaper.

All you have to do is to decide on the figure you want repeating and note down the numbers for the VDU 23 statements of all the user defined characters used.

It's just as we do in our monthly Casting Agency.

The complicated bit is that the program uses 25 user defined characters to make up one figure. It arranges them into one block, using the methods shown in Casting Agency in the November issue of *Electron User*.

It then prints this block over and over again to produce the frieze.

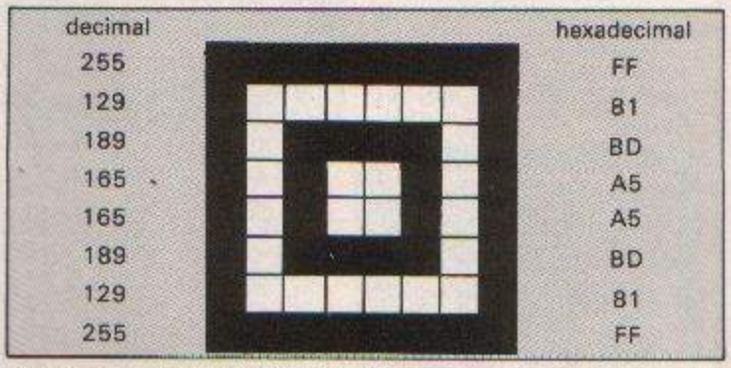


Figure I: Decimal and hexadecimal numbers for the same character

Let's try out a simple pattern. Rather than make up a figure out of 25 user defined characters, we'll just use the same character 25 times over to make up the figure.

Suppose we use the character shown in Figure I. We would define it, just like any other Casting Agency character, with a VDU23 statement. In this case:

```
VDU23,224,255,129,189,
165,165,189,129,255
```

We use this 25 times to create one block. If I was creating a more complicated block, each user defined character would probably be different.

This would mean some planning on a piece of paper beforehand.

Happily the program saves us a lot of time and trouble because it will do all the arranging for us.

What we have to do is put the last eight numbers of the VDU statement into the DATA statements at the end of the program.

Normally we write the numbers after the VDU23,224 in ordinary decimal figures.

However this program makes use of hexadecimal numbers – that is, numbers to the base 16.

Don't worry too much about these. We will be covering hexadecimal numbers in a future Maths Workout feature in *Electron User*.

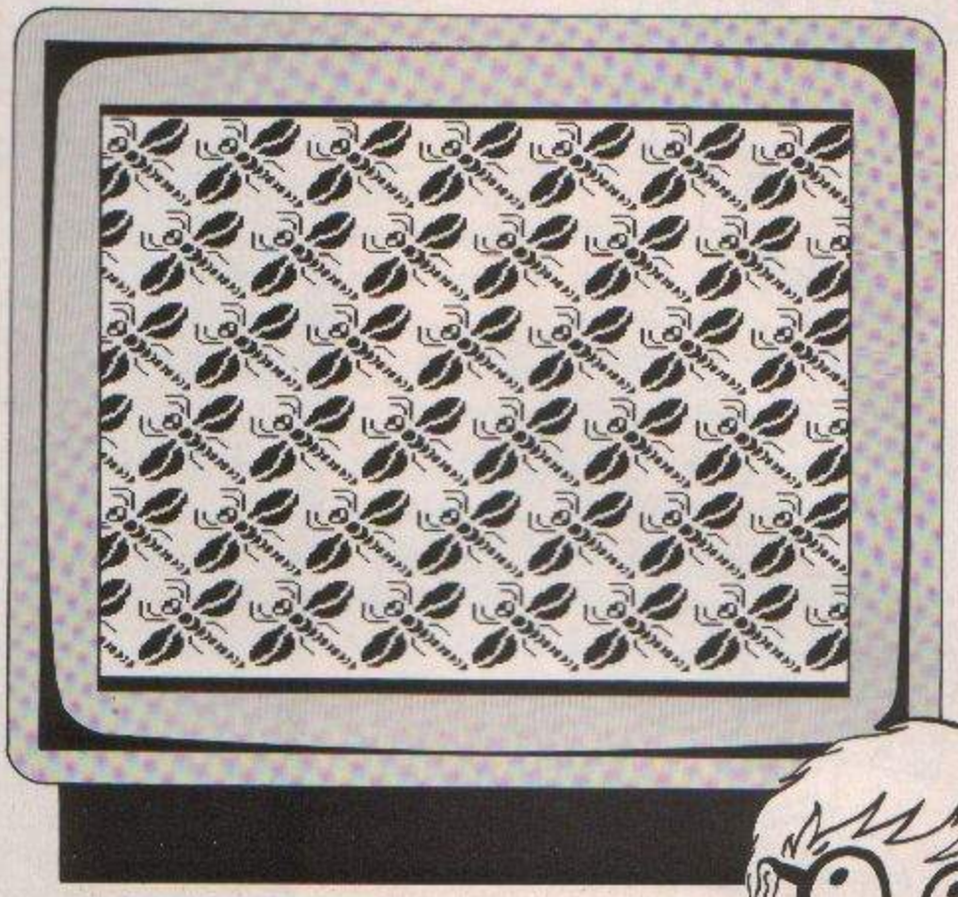
Use Program II to change

This listing was produced using a special formatter which breaks one program line over several lines of listing. When entering a line don't press Return until you come to the next line number. Full details of the formatter is given on Page 4 of the February issue.

```
  1 REM FRIEZE
  2 REM BY ALLEN PLUME
  3 REM (C) ELECTRON USER

  5 *TV0,1
 10 MODE 4
 15 FOR C%=224 TO 248
 20 VDU 23,C%
 25 READ A$
 30 FOR J%=1 TO 15 STEP 2
 35 VDU EVAL ("&"+MID$(A$
    ,J%,2))
 40 NEXT
 45 NEXT
 50 VDU 5
 55 BS$=CHR$ 10+STRING$(5
    ,CHR$ 8)
 60 A$=""
 65 FOR J%=224 TO 244
    STEP 5
 70 FOR I%=J% TO J%+4
 75 A$=A$+CHR$ I%
 80 NEXT
 85 A$=A$+BS$
 90 NEXT
 95 FOR Y%=192 TO 832
    STEP 320
100 FOR X%=0 TO 1240
    STEP 160
105 MOVE X%,Y%
110 PRINT A$
115 NEXT
120 NEXT
125 FOR Y%=352 TO 992
    STEP 320
130 FOR X%=-80 TO 1160
    STEP 160
135 MOVE X%,Y%
140 PRINT A$
145 NEXT
150 NEXT
155 GCOL 0,128
160 VDU 24,0;0;100;1023;16
165 VDU 24,1180;0;1279;1023;16
170 GCOL 0,129
175 VDU 24,100;0;1180;24;16
180 VDU 24,100;1000;1180;1023
    ;16
185 VDU 26
190 MOVE 100,17
    :DRAW 100,1023
195 MOVE 1180,17
    :DRAW 1180,1023
200 VDU 30
205 END
210 DATA 0000000000000000
215 DATA 7F00003C02010101
220 DATA 00B0404040414343
225 DATA 030F1F3FFFFFFFFE
230 DATA E0E0C8C898183870
235 DATA 0080909090908887
240 DATA 1C7272F28E8E7C01
245 DATA 47470F1E1801C3E0
250 DATA F8E1071FFFFFFF7C
255 DATA E0E0E0C0C0800000
260 DATA 403F000000000307
265 DATA 03C3011838F0F2E6
270 DATA E8ECCC1CFD7B070E
275 DATA 0000F00804024140
280 DATA 0000000000000000
285 DATA 0F0F1F3F7F7FFFFE
290 DATA CECFCF9F9F3F3E7E
295 DATA 2023202010080402
300 DATA C0B070640C1B0706
305 DATA 0000000000000040
310 DATA F8F1C30F3E000000
315 DATA FCF8E00000000000
320 DATA 0000000000000000
325 DATA 0001000000000000
330 DATA C0D010640C1C0000
```

This listing is included in this month's cassette tape offer. See order form on Page 43.

Program I



each of the last eight figures into hexadecimal. You then put these odd looking numbers into the relevant DATA statements, one after the other, with no commas.

We will come to this after we've seen how each of the 25 DATA statements at the end of the program correspond to each of the 25 characters that make up the blocks of the frieze.

But first, key in Program II:

```
10 REM PROGRAM II
20 REPEAT
30 PRINT "ENTER THE
   NORMAL NUMBER"
40 INPUT decimal
50 PRINT"THE HEXADECIMAL
   IS ";~decimal
60 UNTIL FALSE
```

Figure II shows how one of these blocks, or figures, is made up of 25 user defined characters.

The top left character of the block – numbered 1 in the diagram – has its VDU data, which are the last eight numbers converted to hexadecimal, stored after the DATA of line 210.

The next, number 2, has its VDU23 numbers stored in line 215 ... and so on until the figures for character 25 are stored in line 330.

In my case, I just want my simple pattern repeated 25 times to form a block, so my DATA statements are all the same, as shown in this listing:

```
210 DATAFF81BDA5A5BD81FF
215 DATAFF81BDA5A5BD81FF
220 DATAFF81BDA5A5BD81FF
225 DATAFF81BDA5A5BD81FF
230 DATAFF81BDA5A5BD81FF
235 DATAFF81BDA5A5BD81FF
240 DATAFF81BDA5A5BD81FF
245 DATAFF81BDA5A5BD81FF
250 DATAFF81BDA5A5BD81FF
255 DATAFF81BDA5A5BD81FF
260 DATAFF81BDA5A5BD81FF
265 DATAFF81BDA5A5BD81FF
270 DATAFF81BDA5A5BD81FF
275 DATAFF81BDA5A5BD81FF
280 DATAFF81BDA5A5BD81FF
285 DATAFF81BDA5A5BD81FF
290 DATAFF81BDA5A5BD81FF
295 DATAFF81BDA5A5BD81FF
300 DATAFF81BDA5A5BD81FF
305 DATAFF81BDA5A5BD81FF
310 DATAFF81BDA5A5BD81FF
315 DATAFF81BDA5A5BD81FF
320 DATAFF81BDA5A5BD81FF
325 DATAFF81BDA5A5BD81FF
330 DATAFF81BDA5A5BD81FF
```

If you still cannot see how the characters fit together to make the blocks, try changing the figures in the DATA statements and see what happens to the patterns.

The last eight numbers of the VDU23,224 making up my character have been converted into hexadecimal, using Program II, and placed in the DATA statements, one after the other without commas.

The 255 becomes FF, 129 becomes 81 and so on. This means that:

255,129,189,165,
165,189,129,255

becomes:

FF81BDA5A5BD81FF

Now when I run the main program with these altered DATA lines I get a brand new pattern. This is far easier to do than describe.

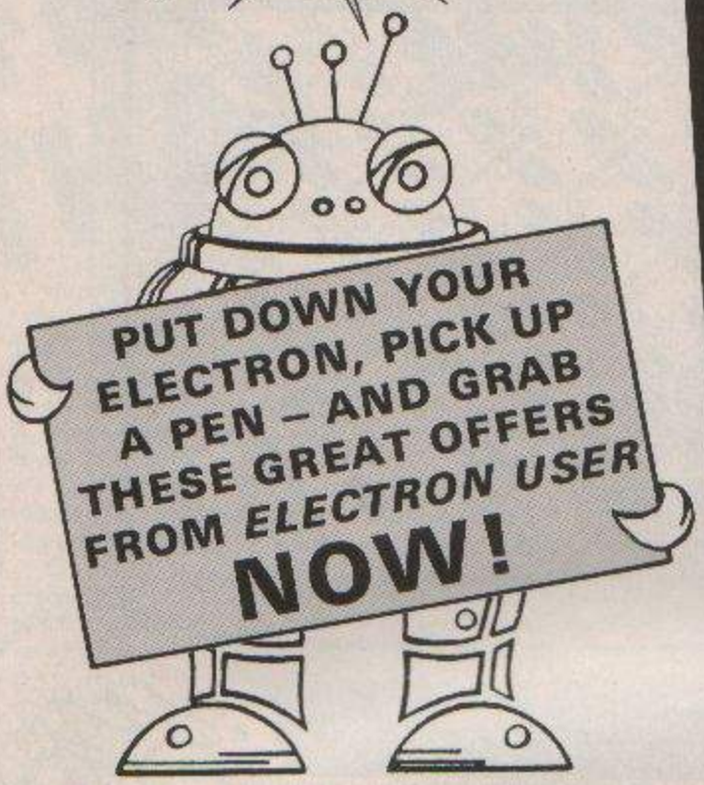
At first, just try your hands at simple patterns like mine. Then as you get more confident try more complicated figures.

It's great fun, and shows just one way in which your Electron can be used as a design tool. I look forward to the results.

1 line 210	2 line 215	3 line 220	4 line 225	5 line 230
6 line 235	7 line 240	8 line 245	9 line 250	10 line 255
11 line 260	12 line 265	13 line 270	14 line 275	15 line 280
16 line 285	17 line 290	18 line 295	19 line 300	20 line 305
21 line 310	22 line 315	23 line 320	24 line 325	25 line 330

Figure II: One block of 25 characters and the lines where their data is stored

electron user



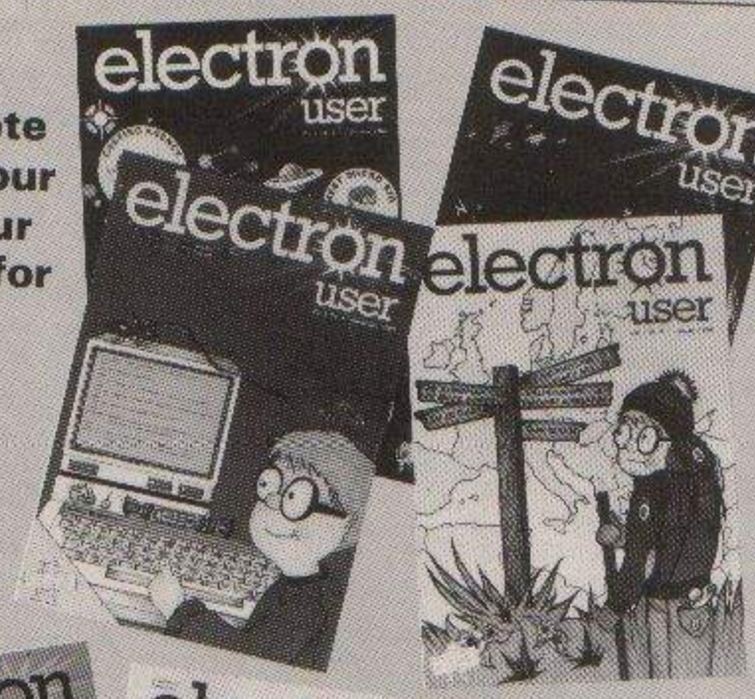
Be one of the first to get each issue

A subscription will ensure you get your own personal copy HOT OFF THE PRESSES month after month for the next year.

Every owner of an Electron – and everyone thinking of buying one – needs to get Electron User every month. It's the brightest, most authoritative yet completely independent guide to a machine that has so much potential you will never tire of reading about its remarkable capabilities.

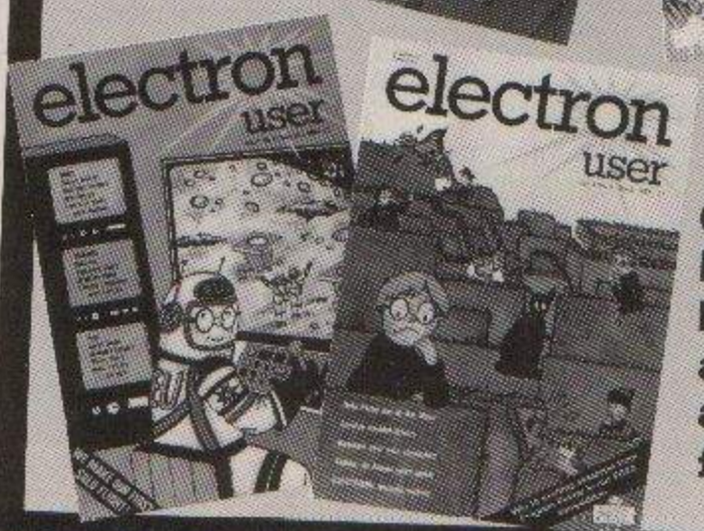
You can buy Electron User from your local newsagent or station bookstall. Or you can take out a 12 months subscription and have it delivered to you by post.

Complete set of our first four issues for only £1.50



Your Electron needs protecting!

Protect your Electron with our luxury dust cover made of soft pliable water-resistant vinyl, bound with strong cotton and decorated with *Electron User* logo. **£3.95**



Copies of February and March issues are still available at £1.25 each

Keep your collection of *Electron User* complete with these handsome binders

Bound in attractive red pvc with the *Electron User* logo in gold blocking on the spine, this binder will hold 12 magazines firmly secured in place by metal rods. **£3.95**

Cassette tapes of *Electron User* programs

Save typing in programs from *Electron User* by sending for these program-packed tapes. **£3.75**

ORDER FORM

All prices include postage, packing and VAT, and are valid to April 27.

Please enter number required in box £ p

Electron User annual subscription

UK £12 ☐
EIRE £13 (IR £16) ☐
Overseas (Surface) £20 ☐
Overseas (Airmail) £40 ☐

Commence with _____ issue TOTAL _____

Electron User introductory issues

£1.50 UK
£1.75 Overseas (Surface)

Complete set of 4 ☐

TOTAL _____

Electron User back issues

£1.25 UK
£1.50 Overseas (Surface)
Airmail prices on application

February ☐
March ☐

TOTAL _____

Electron User tapes

£3.75 (UK & Overseas)

26 Introductory programs ☐
Lunar Lander. February ☐
Chicken. March ☐
Spacehike. April ☐

TOTAL _____

Dust Cover

£3.95 (UK & Overseas)

TOTAL _____

Binder

£3.95 UK
£5.00 Overseas

TOTAL _____

Payment: please indicate method (✓) **TOTAL** _____

☐ Access/Mastercharge/Eurocard
☐ Barclaycard/Visa
☐ American Express

Card No. _____

Expiry Date _____

☐ Cheque/PO made payable to Database Publications Ltd

Name _____

Address _____

_____

Signed _____

Send to: *Electron User*, FREEPOST, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.

(No stamp needed if posted in UK) *Please allows 28 days for delivery*

You can also order by phone

Telephone: 061-480 0171
24 hours

Don't forget to quote your credit card number and give your full address

MCMLXXXIV

If you thought your micro had more to do with the future than the past, let MIKE MAHON show you how to conquer those ancient Roman numerals

YOU may be a whizz at decimal arithmetic or can think in hexadecimal and binary. But how quickly did you work out the title of this article?

The program listing given here will let you do just that – convert Roman numerals to decimal and vice-versa. But more about the program later.

The Romans used a seven-character – septal – system for numeration. These characters and their decimal equivalents are shown in Table I.

Initially the Romans themselves used up to four characters of any one type to make up a number, such as IIII for 4.

But modern usage is based on the subtractive system whereby only three characters of a type are used together and then one is subtracted from the next higher value, like III for 3 and IV for 4. We will be using the subtractive system here.

Did you know that the largest number you can have using this system is 3999? The program described here works in whole numbers – integers – from 1 to 3999.

Do you know that the longest roman numeral is 15 characters long? The answer to this, and some other frequently used figures, is given in Table II.

The program is written in BBC Basic and may be said to be structured in that it is made up of separate modules.

It does not use GOTO or GOSUB or refer to line numbers within it, and the main variables and procedures are reasonably self explanatory.

This should enable the user to readily modify the program for his or her own needs, such as by adding routines for testing and scoring pupils or for printouts.

The main program occupies lines 100-200, most of which is concerned with precautionary features such as switching off the cassette motor and printer, if available, and disabling the auto repeat, cursor editing and copy key functions.

It also forces the program to re-run if either the Break or Escape keys are pressed.

The only way to exit the program and reset all the functions to normal is to press the Control and Break keys together.

The rest of the program is in the procedures, which are listed and explained in Table III.

The main algorithm – the programmed formula – for decimal to Roman conversion is in line 730. This steps through the decimal number and picks out the appropriate roman characters from the data table fed into the array *roman$*.

The other algorithm, for Roman to decimal conversion, is a little longer and resides in lines 1200-1320.

The majority of the program is concerned with – as usual – trapping user errors and presenting information on the screen.

Most problems are catered for and only valid inputs are allowed. It is, however, essential to use the Electron with the Caps lock ON and the Shift lock OFF as at switch on. It is left as an exercise to the reader to find a way around these problems.

Also, what about adding a routine of your own for converting hexadecimal to Roman numerals using the inbuilt facilities of your micro.

Oh – the title of this article? 1984 of course!

Roman	I	V	X	L	C	D	M
Decimal	1	5	10	50	100	500	1000

Table I

Decimal	Comment	Roman
1	Smallest	I
3999	Largest	MMMCMXCIX
3888	Longest	MMMDCCCLXXXVIII
1066	Battle of Hastings	MLXVI
1969	Men on the Moon	MCMLXIX
1983	Birth of *Micro User* & *Electron User*	MCMLXXXIII
2000	The next century	MM

Table II

PROCEDURES	
PROCdata	Fills array *roman$* with all the valid Roman numeral character groups in units, tens, hundreds and thousands.
PROCtitle	Displays the program title and lists the three options available.
PROCdecinput	Receives decimal input for conversion to a Roman numeral.
PROCdecanalyse	Converts decimal input *dec* to equivalent Roman numeral *romnum$*.
PROCromaninput	Receives your Roman numeral *RN$* for conversion to a decimal number.
PROCromananalyse	Converts a Roman numeral to a decimal number.
PROClist	Lists decimal and Roman numbers in the range specified by start and finish.
PROCcheckinput	Allows only the 10 valid decimal and 7 valid Roman characters to be entered.
PROCinvalid	Tells you that your entry is not valid, for example not in the range 1 to 3999.
PROCreturn	Displays message to terminate your input.
PROCpause	Waits for you to have another go to change your option.

Table III

This listing was produced using a special formatter which breaks one program line over several lines of listing. When entering a line don't press Return until you come to the next line number. Full details of the formatter are given on Page 4 of the February issue.

```
100 REM ROMAN NUMERALS
110 REM Mike Mahon
120 REM (C) ELECTRON USER
130 VDU 3
    :*MOTOR 0
135 *FX4,1
140 *KEY10 OLD|M RUN|M
145 *FX11,0
150 MODE 6
160 ON ERROR RUN
170 DIM roman$(4,10)
180 PROCdata
190 PROCtitle
200 END
499
500 DEF PROCdata
510 DATA 0,I,II,III,IV
    ,V,VI,VII,VIII,IX
520 DATA 0,X,XX,XXX,XL
    ,L,LX,LXX,LXXX,XC
530 DATA 0,C,CC,CCC,CD
    ,D,DC,DCC,DCCC,CM
540 DATA 0,M,MM,MMM,0
    ,0,0,0,0,0
550 FOR row%=1 TO 4
    :FOR col%= 0 TO 9
    :READ roman$(row%
    ,col%)
    :NEXT
    :NEXT
560 ENDPROC
599
600 DEF PROCdecinput
605 okay$="1234567890"
    :maxlen=4
610 CLS
    :PROCreturn
    :PRINT TAB(2,5)"Enter
     Decimal number  ";
    :PROCcheckinput
    :dec=VAL (string$)
620 IF dec <1 OR dec>3999
     OR dec<>INT (dec)

    THEN PROCinvalid
    :PROCdecinput
630 PROCdecanalyse
640 PRINT TAB(8,8)"Roman
     numeral   ";romnum$
650 PROCpause
    :PROCdecinput
660 ENDPROC
699
700 DEF PROCdecanalyse
710 dec$=STR$ (dec)
    :pos=0
    :romchar$=""
    :romnum$=""
720 FOR row%= LEN (dec$)
     TO 1 STEP -1
725 pos=pos+1
730 romchar$=roman$(row%
    ,VAL (MID$(dec$,pos
    ,1)))
735 IF romchar$="0"
    THEN romchar$=""
740 romnum$=romnum$+romchar$
750 NEXT
760 ENDPROC
799
800 DEF PROCromaninput
810 okay$="IVXLCDM"
    :maxlen=15
820 CLS
    :PROCreturn
    :PRINT TAB(2,5)"Enter
     Roman numeral   ";
    :PROCcheckinput
    :RN$=string$
850 PROCromananalyse
860 IF DEC<1 OR DEC>3999

    THEN PROCinvalid
    :PROCromaninput
865 dec=DEC
    :PROCdecanalyse
    :IF RN$ <> romnum$

    THEN PROCinvalid
    :PROCromaninput
870 PRINT TAB(8,8)"Decimal
     number  ";DEC
880 PROCpause
    :PROCromaninput
890 ENDPROC
899
900 DEF PROCpause
905 *FX15,0
910 PRINT TAB(1,22)"Press
     ESCAPE for MENU  RETUR
    N for more"
920 REPEAT
    :key=GET
    :UNTIL key=13
930 ENDPROC
999
1000 DEF PROClist
1005 okay$="1234567890"
    :maxlen=4
1010 CLS
    :PROCreturn
    :PRINT TAB(2,2)"Enter
     range (decimal): START
      ";
    :PROCcheckinput
    :start=VAL (string$)
    :PRINT TAB(25,3)"FINISH
      ";
    :PROCcheckinput
    :finish=VAL (string$)
1020 IF start<1 OR start>399
    9 OR finish<1 OR finish
    >3999 OR finish<start

    THEN PROCinvalid
    :PROClist
1030 start=INT (start)
    :finish=INT (finish)
1035 IF finish-start >15

    THEN VDU 14
    :PRINT TAB(1,22)"Press
     SHIFT to Scroll page
          "
1040 VDU 28,5,20,38,5
1050 FOR dec=start TO finish
    :PROCdecanalyse
1060 PRINT TAB(5);dec;
    TAB(15);romnum$
1070 NEXT
1080 VDU 26
    :VDU 15
1090 PROCpause
    :PROClist
1100 ENDPROC
1199
1200 DEF PROCromananalyse
1210 L=LEN (RN$)
    :DEC=0
1220 N=0
    :REPEAT
    :N=N+1
    :IF MID$(RN$,L,1)=
    "I"
    THEN DEC=DEC+1
    :L=L-1
    :UNTIL L=0 OR N=3
1230 IF MID$(RN$,L,1)="V"

    THEN DEC=DEC+5
    :L=L-1
    :IF MID$(RN$,L,1)=
    "I"
    THEN DEC=DEC-1
    :L=L-1
1240 N=0
    :REPEAT
    :N=N+1
    :IF MID$(RN$,L,1)=
    "X"
    THEN DEC=DEC+10
    :L=L-1
    :IF MID$(RN$,L,1)=
    "I"
    THEN DEC=DEC-1
    :L=L-1
    :N=N-1
1250 UNTIL L=0 OR N=3
1260 IF MID$(RN$,L,1)="L"

    THEN DEC=DEC+50
    :L=L-1
    :IF MID$(RN$,L,1)=
    "X"
    THEN DEC=DEC-10
    :L=L-1
1270 N=0
    :REPEAT
    :N=N+1
    :IF MID$(RN$,L,1)=
    "C"
    THEN DEC=DEC+100
    :L=L-1
    :IF MID$(RN$,L,1)=
    "X"
    THEN DEC=DEC-10
    :L=L-1
    :N=N-1
1280 UNTIL L=0 OR N=3
1290 IF MID$(RN$,L,1)="D"
```

Roman Numerals listing

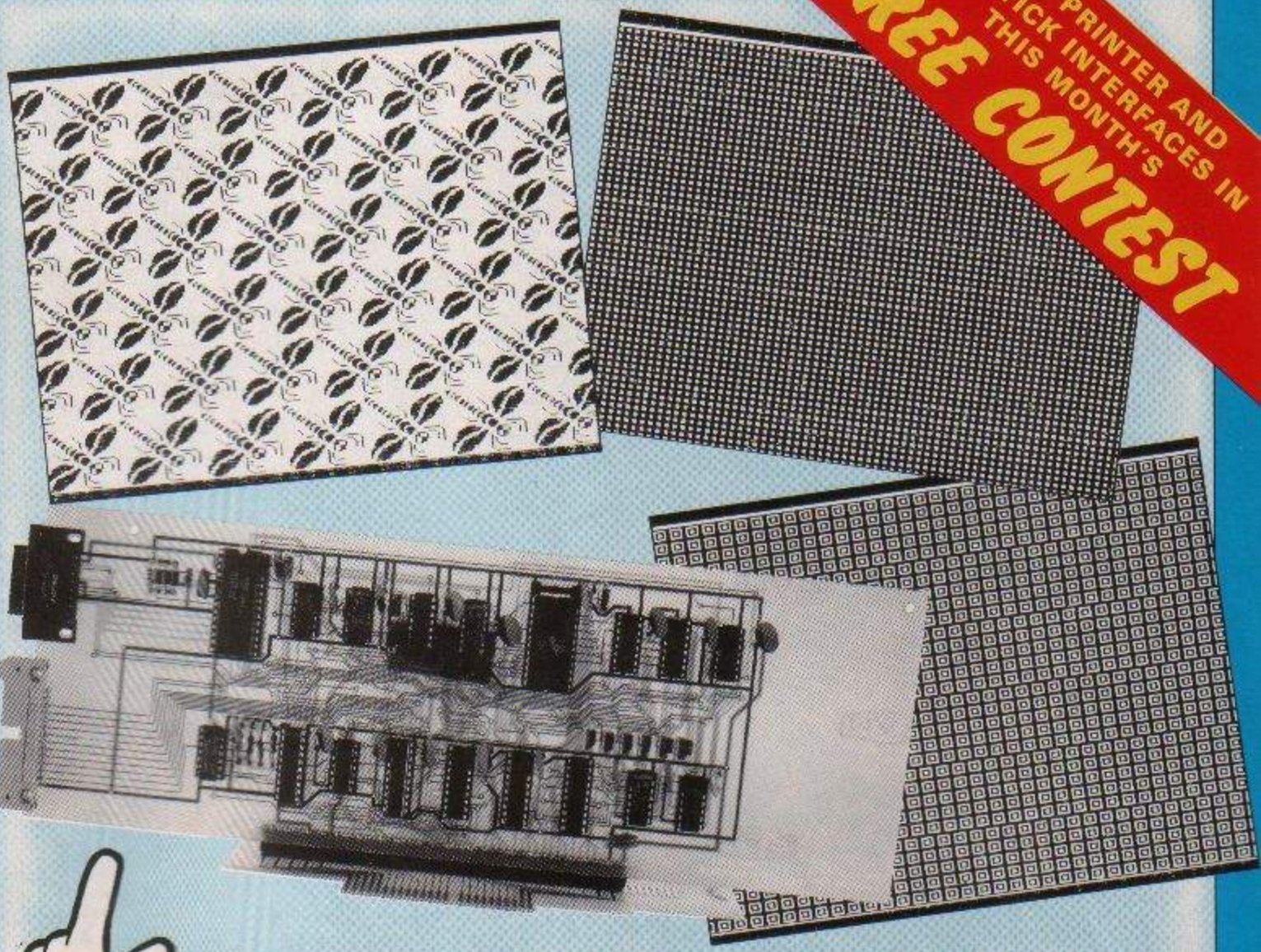
From Page 45

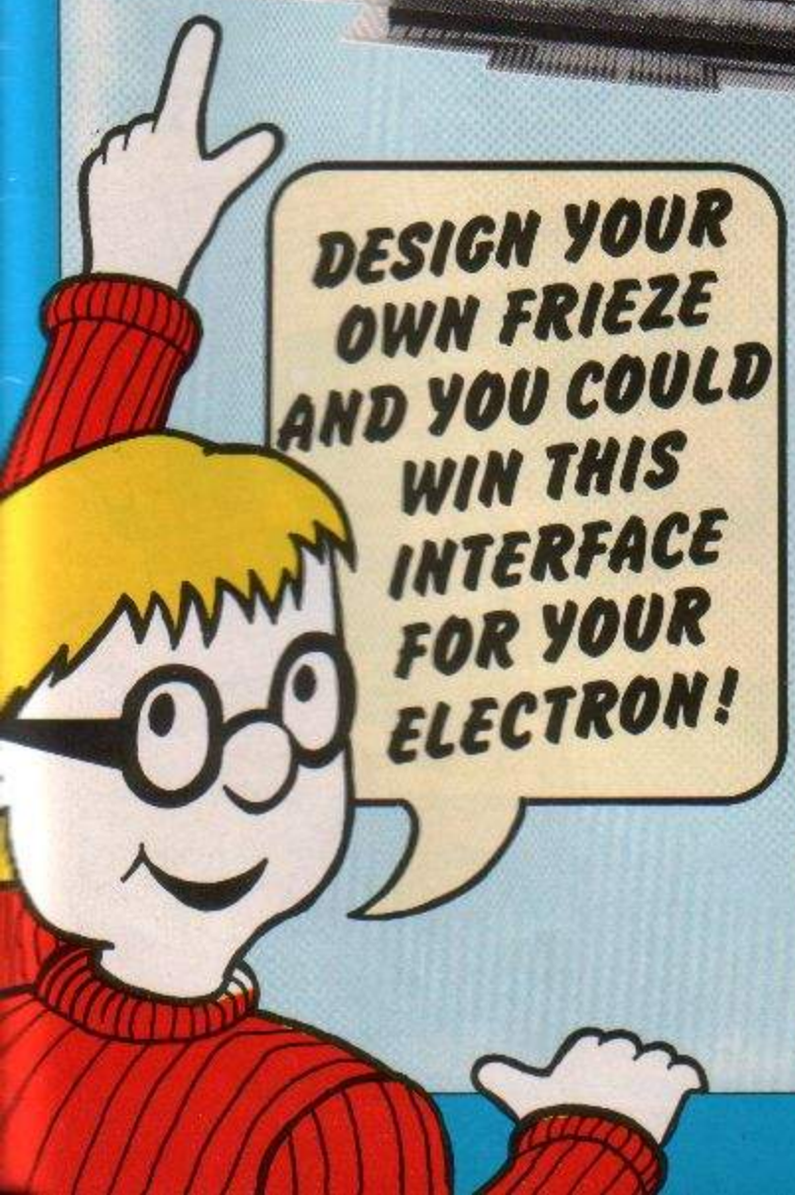
```
    THEN DEC=DEC+500
    :L=L-1
    :IF MID$(RN$,L,1)=
    "C"
    THEN DEC=DEC-100
    :L=L-1
1300 N=0
    :REPEAT
    :N=N+1
    :IF MID$(RN$,L,1)=
    "M"
    THEN DEC=DEC+1000
    :L=L-1
    :IF MID$(RN$,L,1)=
    "C"
    THEN DEC=DEC-100
    :L=L-1
    :N=N-1
1310 UNTIL L=0 OR N=3
1320 ENDPROC
1999
2000 DEF PROCtitle
2010 CLS
2015 VDU 19,1,3,0,0,0
2020 PRINT TAB(10,5)"  ROMAN
     NUMERALS"
2040 PRINT TAB(8,10)"1
     Decimal to Roman"
2050 PRINT TAB(8,12)"2
     Roman to Decimal"
2060 PRINT TAB(8,14)"3
     Listing of Roman"
2070 VDU 19,1,2,0,0,0
    :PRINT TAB(0,17)"
       Select appropriate
     option ";
2080 REPEAT
    :option$=GET$
    :UNTIL option$="1"
     OR option$="2"
    OR option$="3"
2085 VDU 19,1,7,0,0,0
2090 IF option$="1"
    THEN PROCdecinput
2100 IF option$="2"
    THEN PROCromaninput
2110 IF option$="3"
    THEN PROClist
2130 ENDPROC
2199
2200 DEF PROCinvalid
2210 VDU 7,7
    :CLS
    :VDU 19,1,11,0,0,0
    :PRINT TAB(14,10)"INVAL
    ID ENTRY"
2220 TIME =0
    :REPEAT
    :UNTIL TIME =200
2230 VDU 20
    :ENDPROC
2299
2300 DEF PROCcheckinput
2315 string$=""
    :REPEAT
2320 REPEAT
    :key$=GET$
    :UNTIL INSTR(okay$
    ,key$) >0 OR key$=
     CHR$ (13)
2330 PRINT key$;
    :IF key$ <> CHR$ (13)
    THEN string$=string$+ke
    y$
2340 UNTIL key$= CHR$ (13)
     OR LEN (string$)
     >= maxlen
2350 ENDPROC
2399
2400 DEF PROCreturn
2410 PRINT TAB(1,22)"Press
     RETURN to input entry"
    :ENDPROC
```

This listing is included in this month's cassette tape offer. See order form on Page 43.

WIN PRINTER AND JOYSTICK INTERFACES IN THIS MONTH'S

FREE CONTEST





Here's your chance to add more power to your Electron with our latest free competition. It's easy to enter and tests both your Electron know how and your artistic abilities.

The prize is one of Sir Computers' printer and joystick interfaces for your Electron.

All you have to do is design your own frieze using the frieze program from this issue. When you've decided on a figure you think makes a nice pattern, just send us a diagram of it along with the VDU23s that form its user defined characters.

You don't even have to send us a cassette of the program, just the diagram and VDU23s will do.

The most original and artistic frieze we receive will win the printer and joystick interface. The closing date is April 28 and the judges decision is final.

ELECTRON USER CONTEST

Attach this coupon to your entry.

Attached is my entry for the frieze competition.

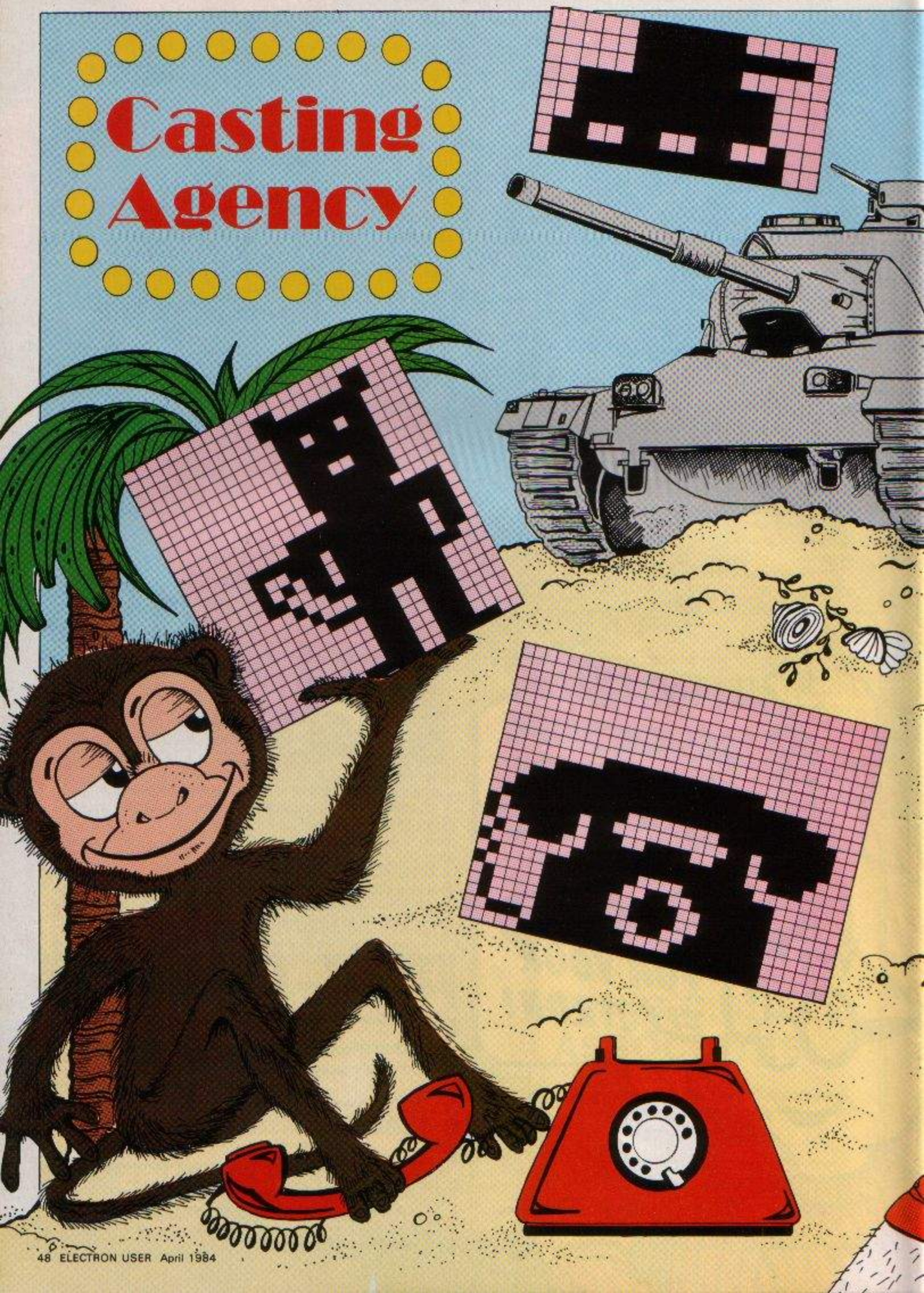
NAME .....

ADDRESS .....

.....

Post to: FRIEZE, *Electron User* Contest, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.

Casting Agency

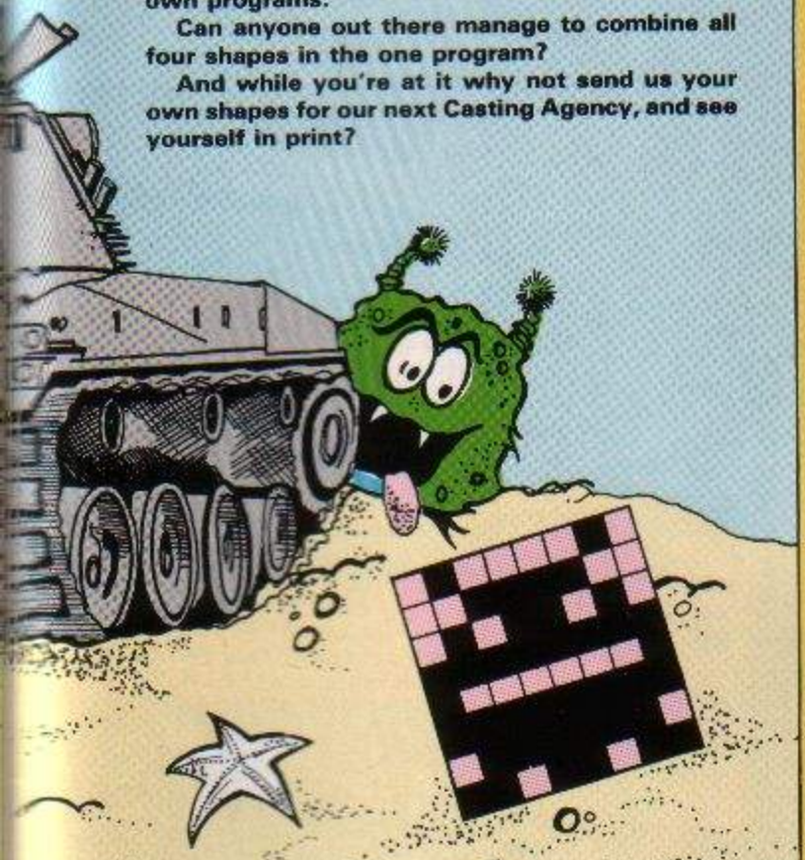


FOUR more characters from our Casting Agency for you. This month, as well as a simple muncher, we've included three compound characters – a rabbit, a telephone and a tank.

As usual, we're giving you the VDU 23 statements so you can use the characters in your own programs.

Can anyone out there manage to combine all four shapes in the one program?

And while you're at it why not send us your own shapes for our next Casting Agency, and see yourself in print?



MUNCHER

From Jonathan Stone (Skegness)

```
VDU 23,240,66,60,90,
255,129,255,126,219
```

TANK

From S. Murray (Leek Wootton)

```
VDU 23,240,7,7,7,
63,63,63,102,127

VDU 23,241,224,254,224,
248,248,248,100,252
```

MONKEY

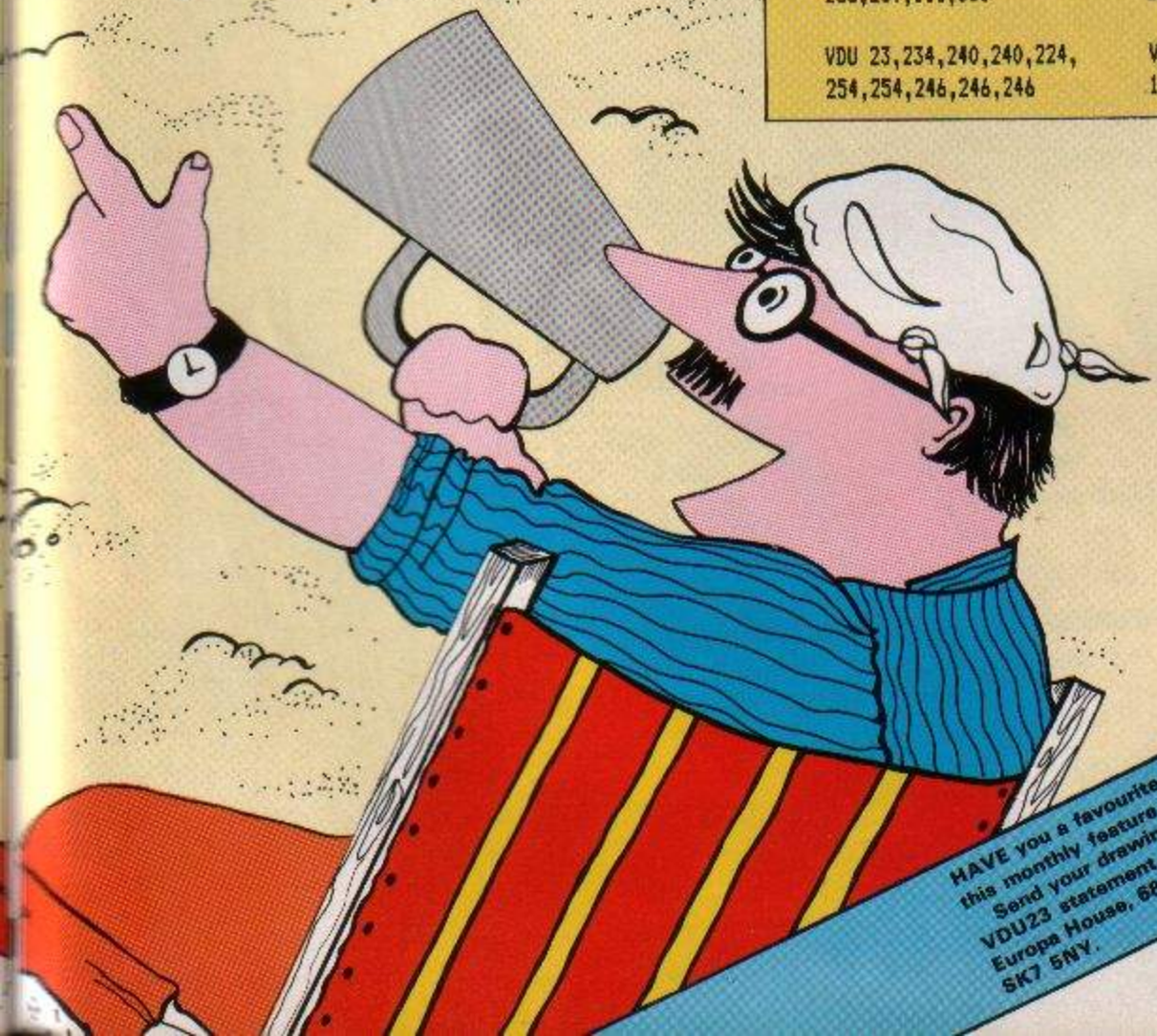
From Gary Peterson (South Shields)

```
VDU 23,230,0,24,31,
15,13,15,15,14

VDU 23,231,0,24,248,
240,176,240,240,112

VDU 23,232,0,0,0,0,
7,4,5,6

VDU 23,233,15,15,7,127,
255,239,111,111

VDU 23,234,240,240,224,
254,254,246,246,246

VDU 23,235,1,0,0,0,
0,0,0,0

VDU 23,236,63,143,126,
14,14,14,14,62

VDU 23,237,252,240,112,
112,112,112,112,124
```

TELEPHONE

From Mark Osborne (Bromley)

```
VDU 23,230,0,7,31,63,
127,191,159,78

VDU 23,231,127,255,255,
255,240,96,115,127

VDU 23,232,254,255,255,
255,15,6,206,254

VDU 23,233,0,224,248,252,
252,252,248,112

VDU 23,234,128,67,131,
67,67,67,59,7

VDU 23,235,255,252,249,
243,243,249,252,255

VDU 23,236,255,63,159,
207,207,159,63,255

VDU 23,237,0,192,192
192,192,192,192,192
```



HAVE you a favourite character you would like to see in this monthly feature in Electron User? Send your drawing of the character, together with the VDU23 statement, to: Shape Dictionery, Electron User, Europa House, 68 Chester Road, Hazel Grove, Stockport SK7 5NY.

SPACE HiKE

SPACEHIKE is an arcade type game loosely based on the arcade classic Frogger but with some new and interesting graphics.

The object is to get four spacemen back to their home base at the top of the screen.

They have to avoid monsters on the bottom four rows, take a rest, and then hitch a ride on various spaceships in the next rows to jump home.

It's quite safe to land anywhere on a spaceship. You don't get killed if you are on the first or last block – unlike in many of the professional games. This makes it slightly easier for younger players.

You have three lives to accomplish your journey. When all four home bases are filled you move on to a harder level.

Level 10 is the most difficult, and if you get through it you are given suitable congratulations.

The levels become harder by blocking in the rest area from the edges and also by increasing the speed of the game.

Every time your score increases by 1000 you get an extra life. Ten points are scored for each jump up.

When all your lives are used up the screen is cleared and the hi score, your score and the level are displayed.

You then have the option of another game, and also that of sound or silent running. This can be an advantage if you want to play in a crowded room.

If you choose silent running you lose a rather nice jingle every time your spaceman reaches home base and other appropriate sounds throughout the game.

HINTS ON TYPING IN:

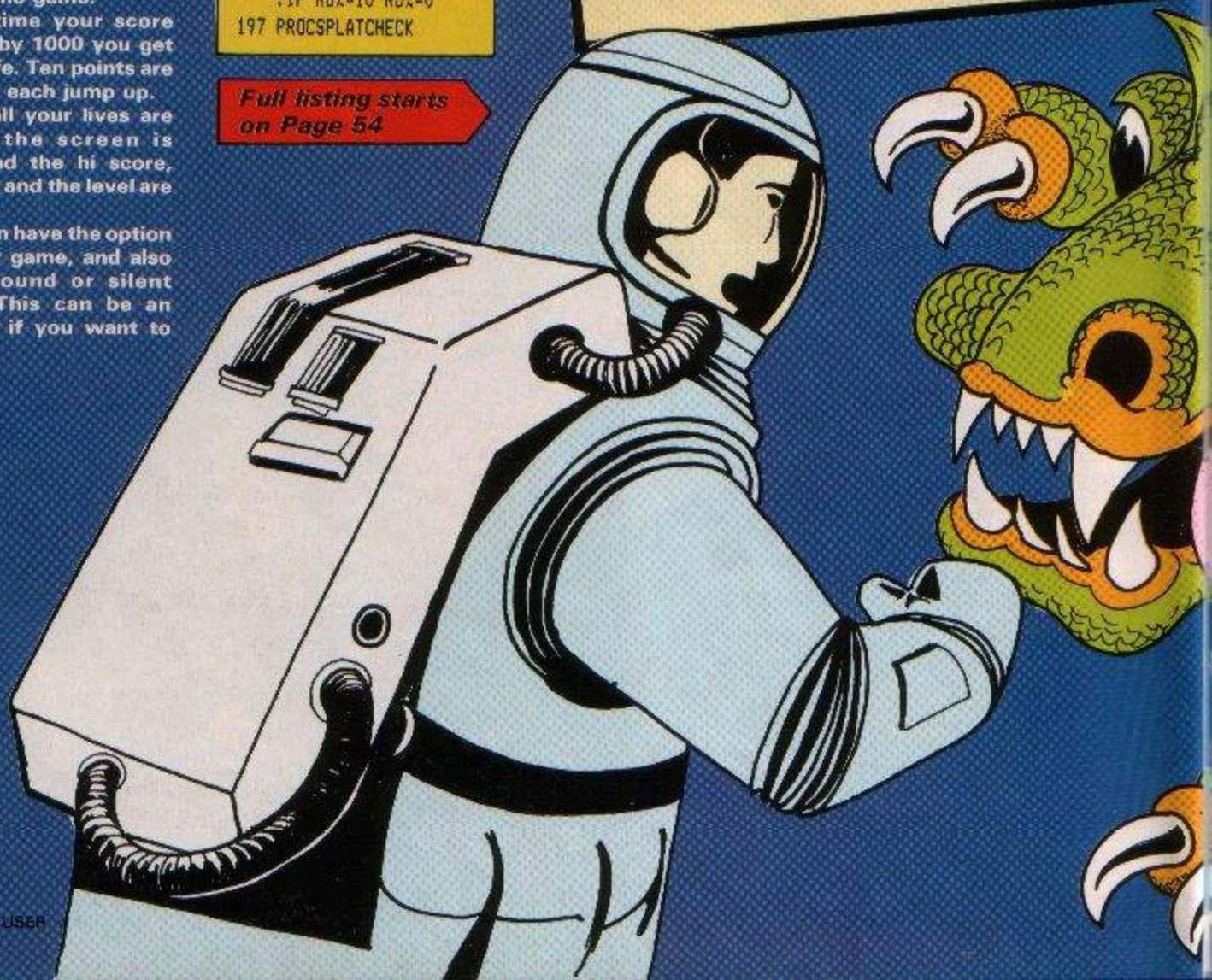
- *Omit line 10 until all errors are found, as this disables the Escape key.*
- *If you want to increase the speed omit line 360.*
- *If you want a harder game add two extra lines:*

```
192 PROCMOVESPACRAFT(RO%)
    :RO%=RO%+1
    :IF RO%=10 RO%=0
197 PROCSPLATCHECK
```

Full listing starts on Page 54

Major procedures

PROCINIT	Initialises variables
PROCSCREEN	Draws screen
PROCMOVEYOU	Moves the man
PROCMOVESPACRAFT(N)	Moves road N in the right direction
PROCSPLATCHECK	Checks whether or not you are dead
PROCDEAD	Kills you in a suitable way
PROCEND	Displays score, hi-score and your level
PROCWELLDONE	Congratulates you appropriately if you beat Level 10
PROCHOME	Checks whether you have jumped into an empty hole
PROCDELETE	Fills in where you were last
PROCSPA(N)	Prints A$(N) at the right position
PROCMOVEYONSPA	Moves you on the spacecraft



Numeric variables

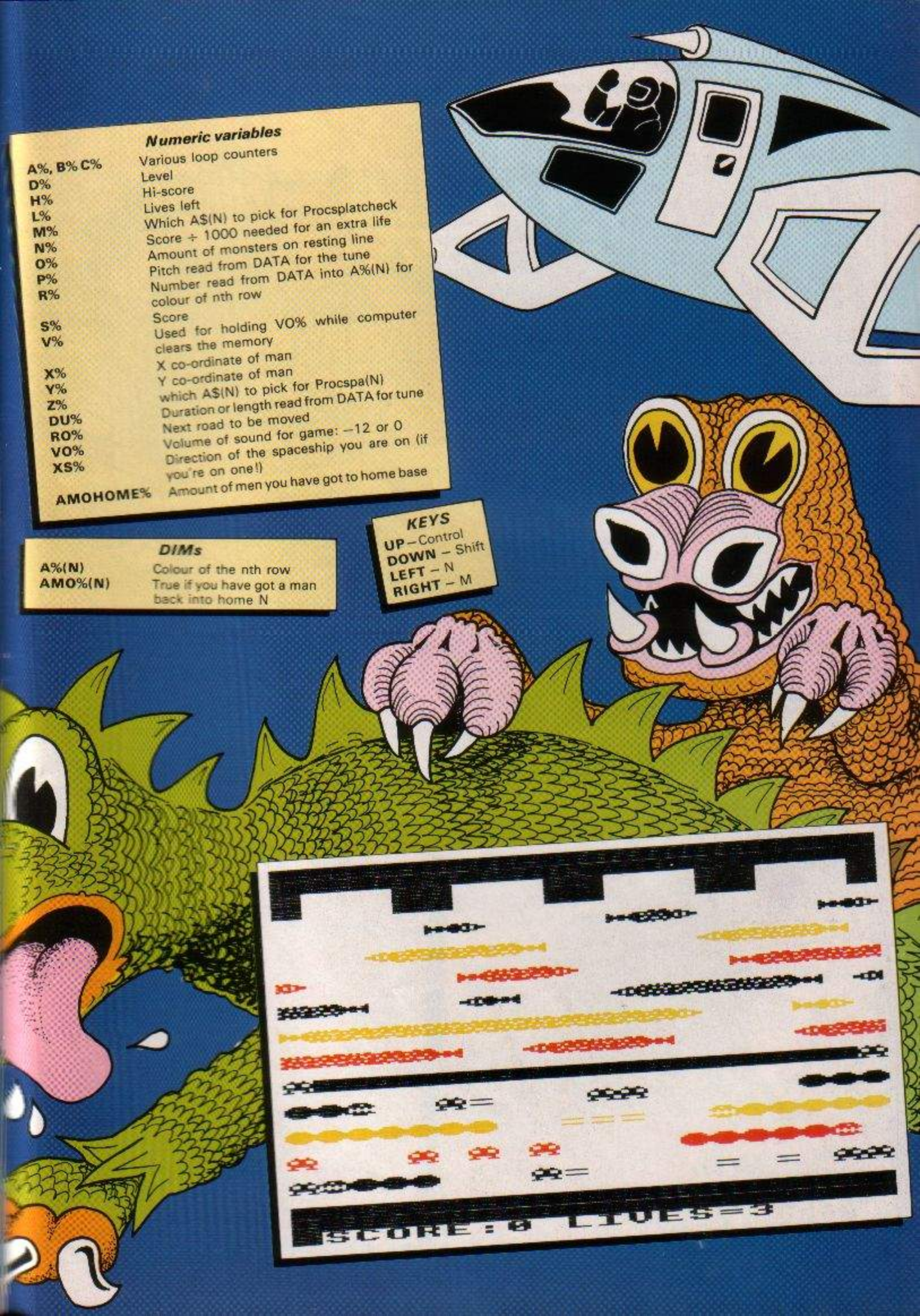
A%, B% C%	Various loop counters
D%	Level
H%	Hi-score
L%	Lives left
M%	Which A$(N) to pick for Procsplatcheck
N%	Score ÷ 1000 needed for an extra life
O%	Amount of monsters on resting line
P%	Pitch read from DATA for the tune
R%	Number read from DATA into A%(N) for colour of nth row
S%	Score
V%	Used for holding VO% while computer clears the memory
X%	X co-ordinate of man
Y%	Y co-ordinate of man
Z%	which A$(N) to pick for Procspa(N)
DU%	Duration or length read from DATA for tune
RO%	Next road to be moved
VO%	Volume of sound for game: −12 or 0
XS%	Direction of the spaceship you are on (if you're on one!)
AMOHOME%	Amount of men you have got to home base

DIMs

A%(N)	Colour of the nth row
AMO%(N)	True if you have got a man back into home N

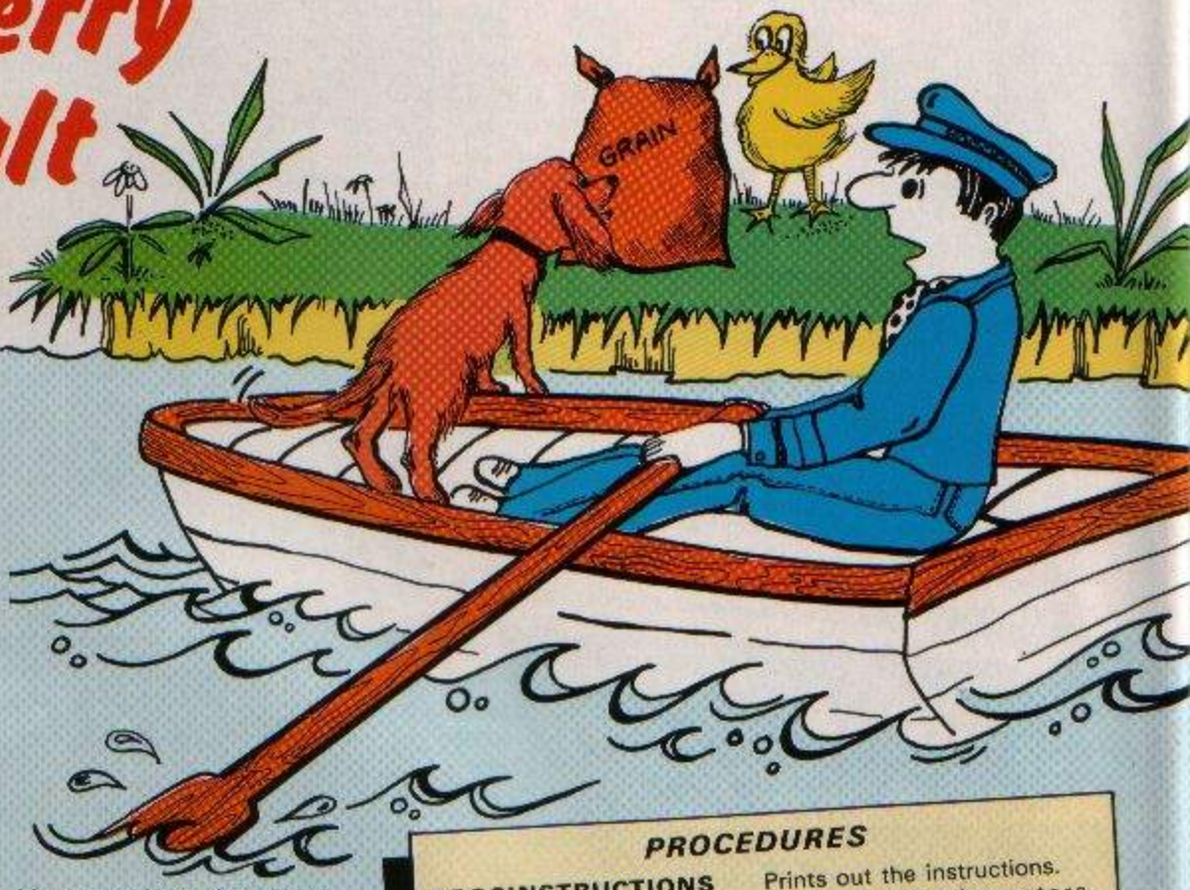
KEYS

UP – Control
DOWN – Shift
LEFT – N
RIGHT – M



Get the dog, duck and grain across the river – but this game by PETE DAVIDSON isn't as simple as it looks. In fact . . .

It's a ferry difficult task!



REMEMBER the old paper and pencil puzzle "Dog, Duck and Grain"? Well, here's the Electron version to tease and test you.

You have to ferry the animals and the grain across the river. But the problem is that once you get in the boat there's only enough room left to carry one of the three.

If you leave the dog behind with the duck, then the duck becomes the dog's dinner. If you leave the duck alone with the grain, then the grain soon becomes the duck's dinner.

And you've got to get them all across the river before you can have your dinner!

Can you do it before you're reduced to eating the duck yourself? How many goes will it take you to solve the Electron "Dog, Duck and Grain" puzzle?

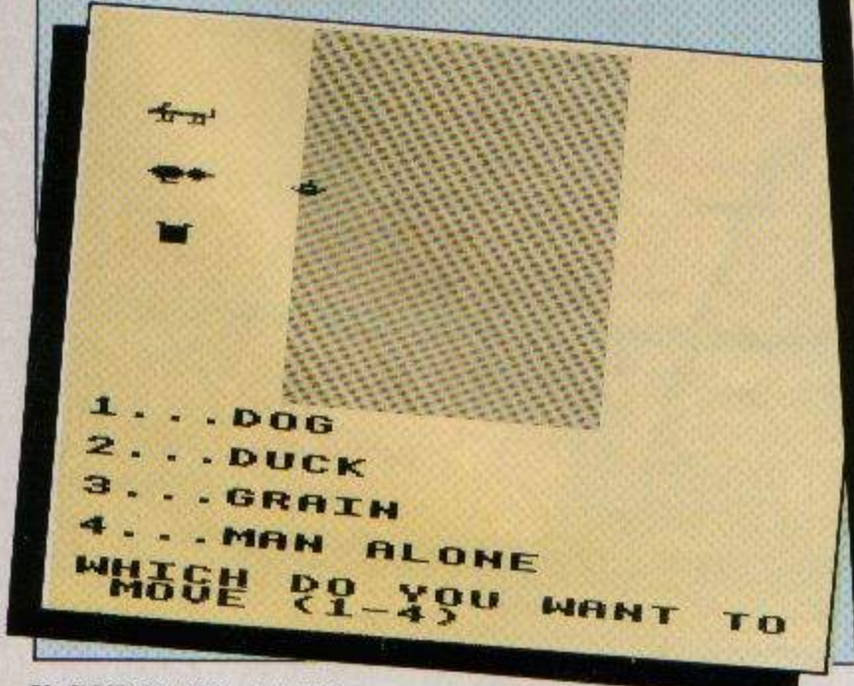
Two variables (SIDE(0)) and SIDE(1)) contain a number from 0 to 7 representing the objects on the left hand side (0) and right hand side (1) of the river.

If they are thought of as binary numbers, then the bits represent the grain, the duck and the dog.

For example, SIDE(0)=7 (111 in binary) means that all three are on the left.

SIDE(0)=5 (101) and SIDE(1)=2 (010) means the dog and grain are on the left, with the duck on the right.

By using the logical operators AND, OR and EOR it is possible to check what is on any side, and remove or put in objects.



PROCEDURES

PROCINSTRUCTIONS	Prints out the instructions.
PROCINIT	Defines the characters, initialises SIDE(0) and SIDE(1) (the objects on each side), RESULT (0 unless you lose), and TRY (the number of times you cross the river). The procedure also draws the initial picture on the screen.
PROCWHICH(p)	The parameter p is the value of SIDE(0) or SIDE(1) and this procedure selects the object you want to move and checks it is actually there.
PROCCHECK(P)	Checks the combination of objects left on the side determined by P (0 is left and 1 is right).
PROCMOVERIGHT	This procedure removes the selected object from the left (changes SIDE(0)) and puts it on the right. It calls PROCDISPLAY at appropriate times to show the positions of the objects on the screen. It also calls PROCSHIFT to show the boat moving.
PROCMOVELEFT	Works like PROCMOVERIGHT to remove objects from the right bank and put them on the left bank.
PROCDISPLAY(p1,p2)	Writes the objects determined by p1 in column p2 on the screen. For example, PROCDISPLAY(7,0) writes all three on the left of the screen.
PROCSHIFT(a,b)	Moves the boat from a to b.
PROCRESULTS	Prints out the results.

This listing was produced using a special formatter which breaks one program line over several lines of listing. When entering a line don't press Return until you come to the next line number. Full details of the formatter are given on Page 4 of the February issue.

```
10 REM DOG,DUCK,GRAIN
20 REM (C) ELECTRON USER
30 MODE 6
   :PROCINSTRUCTIONS
40 DIM SIDE(2)
   :MODE 2
50 PROCINIT
60 REPEAT
70 PROCWHICH(SIDE(0))
80 PROCMOVERIGHT
90 PROCCHECK(0)
100 SOUND 0,0,0,1
110 IF RESULT<>0OR SIDE(1)=7
    THEN 150
120 PROCWHICH(SIDE(1))
130 PROCMOVELEFT
140 PROCCHECK(1)
150 UNTIL SIDE(1)=7OR RESULT<
    >0
160 PROCRESULTS
170 COLOUR 5
180 PRINT '"DO YOU WANT ANOTH
    ER"''"GO?"
    :REPLY$=GET$
    :IF REPLY$="Y"
    THEN RUN
    ELSE IF REPLY$<>"N"
    THEN VDU 7
    :GOTO 180
    ELSE MODE 6
    :END
190 DEF PROCINIT
200 VDU 23;8202;0;0;0;
210 VDU 23,224,0,0,8,8,60
    ,165,126,60
220 VDU 23,225,24,24,48
    ,255,255,20,20,60
230 VDU 23,226,0,2,2,250
    ,254,40,40,120
240 VDU 23,227,0,31,63,127
    ,63,31,8,15
250 VDU 23,228,0,24,188
    ,255,188,24,0,0
260 VDU 23,229,12,7,7,7
    ,7,7,7,7
270 VDU 23,230,48,224,224
    ,224,224,224,224,224
280 ENVELOPE 2,2,6,0,0,255
    ,0,0,126,0,0,-126,126
    ,126
290 SIDE(0)=7
    :SIDE(1)=0
300 RESULT=0
    :TRYS=0
310 GCOL 0,4
320 MOVE 350,400
330 PLOT 1,0,600
340 PLOT 81,560,0
350 PLOT 1,0,-600
360 PLOT 81,-560,0
370 GCOL 3,2
380 PROCDISPLAY(7,0)
390 VDU 5
    :MOVE 328,764
    :VDU 224,4
400 ENDPROC
410 DEF PROCWHICH(SIDE)
420 COLOUR 6
    :TRYS=TRYS+1
430 PRINT TAB(0,20)"1...DOG"'
    '"2...DUCK"''"3...GRAIN"'
    '"4...MAN ALONE"
440 COLOUR 5
450 PRINT TAB(0,28)"WHICH
    DO YOU WANT TO MOVE
    (1-4)"
    :*FX15,1
460 OBJECT=GET -49
    :IF OBJECT<0OR OBJECT>3
    THEN VDU 7
    :GOTO 460
470 OBJECT=2^(OBJECT)
    AND 7
    :IF (OBJECT AND SIDE)<>OB
    JECT
    THEN VDU 7
    :GOTO 460
480 PRINT TAB(0,28)SPC (35)
490 ENDPROC
500 DEF PROCMOVERIGHT
510 SIDE(0)=SIDE(0)EOR OBJECT
    :PROCDISPLAY(SIDE(0)
    ,0)
    :SIDE(1)=SIDE(1)OR OBJECT
520 PROCSHIFT(328,856)
530 PROCDISPLAY(SIDE(1)
    ,15)
540 ENDPROC
550 DEF PROCCHECK(POSITION)
560 IF SIDE(POSITION)=7
    AND POSITION=0
    THEN RESULT=1
570 IF SIDE(POSITION)=3
    THEN RESULT=1
580 IF SIDE(POSITION)=6
    THEN RESULT=2
590 ENDPROC
600 DEF PROCMOVELEFT
610 SIDE(1)=SIDE(1)EOR OBJECT
    :PROCDISPLAY(SIDE(1)
    ,15)
    :SIDE(0)=SIDE(0)OR OBJECT
620 PROCSHIFT(864,336)
630 PROCDISPLAY(SIDE(0)
    ,0)
640 ENDPROC
650 DEF PROCRESULTS
660 PRINT TAB(0,20)SPC (180)
    TAB(0,20);
670 COLOUR 1
680 IF RESULT=0
    THEN PRINT "WELL DONE"
    ELSE ?20
690 SOUND 1,2,4,50
700 PRINT '"YOU CROSSED "
    STR$ (TRYS)" TIMES"
    :IF TRYS=7PRINT '"THE
     BEST POSSIBLE!!"
    ELSE PRINT '"IT IS POSSIB
    LE IN 7"
    :ENDPROC
710 ENDPROC
720 COLOUR 2
730 IF RESULT=2PRINT "THE
     DUCK ATE THE"''"GRAIN"
740 IF RESULT=1PRINT "THE
     DOG ATE THE"''"DUCK"
750 SOUND 0,-15,2,10
760 ENDPROC
770 DEF PROCSHIFT(START
    ,FINISH)
780 VDU 5
790 IF START>FINISH GAP=-8
    :
    ELSE GAP=8
800 FOR I=START TO FINISH
    STEP GAP
810 SOUND 0,-15,6,1
820 SOUND 0,0,0,2
830 MOVE I,764
    :VDU 224
840 MOVE I+GAP,764
850 *FX19
860 VDU 224
870 NEXT
880 VDU 4
890 ENDPROC
900 DEF PROCDISPLAY(OBJECT
    ,POSITION)
910 FOR I%=0TO 2
920 COLOUR I%+1
930 PRINT TAB(POSITION,2+(I%+
    1)*3);
    :IF (OBJECT AND 2^I%)=(2^
    I%)VDU 32,225+I%*2,226+I%
    *2
    ELSE PRINT SPC (5)
940 NEXT
950 ENDPROC
960 DEF PROCINSTRUCTIONS
970 PRINT 'SPC (5)"THE DOG,
    DUCK, & CORN PUZZLE"'
    SPC (5)STRING$(28,"*")
980 PRINT '''Ferry the dog,
    the duck, and the corn"'
    '"to the other side of
    the river."''"At no
    time must you leave
    the dog alone"''"with
    the duck, or the duck
    with the"'"corn."
990 PRINT ''SPC (8)"PRESS
    ANY KEY"
    :*FX15,1
1000 A=GET
1010 ENDPROC
```

This listing is included in this month's cassette tape offer. See order form on Page 43

Space Hike listing

From Page 51

This listing was produced using a special formatter which breaks one program line over several lines of listing. When entering a line don't press Return until you come to the next line number. Full details of the formatter are given on Page 4 of the February issue.

```
 5 REM (C) ELECTRON USER
10 *FX14,6
20 ON ERROR MODE 6
   :REPORT
   :PRINT " at line ";
   ERL
   :END
30 VO%=-12
   :H%=0
40 V%=VO%
   :CLEAR
   :MODE 1
   :VO%=V%
   :COLOUR 2
   :PRINT '"
       SPACE HIKE"'
   :PRINT "      BY MARTIN
    HOLLIS"'
50 PRINT "    FOR THE
    ACORN electron
    "'
   :COLOUR 1
60 PRINT "  The object
   is to fill the top
   holes     by going
   past the four  moving
   rows     of monsters
   and then hitching
   rides     on the six
   rows of space ships
   only     to jump into
   an empty hole right
   at     the top .When
   all four ";
70 PRINT "holes at the
       top are filled
   , you then pass onto
       a harder level"
80 PRINT '"     Good luck!
   !!"'
90 PRINT "     KEYS"
   :PRINT "CTRL.....
   = UP"'"SHIFT.....
   = DOWN"'"N.........
   = LEFT"'"M.........
   = RIGHT"
100 PRINT "Press any key
   to start or Q for
   quiet orS for sound";
   :G$=GET$
   :IF G$="Q" OR G$="q"

   THEN VO%=0
   ELSE IF G$="S"
   OR G$="s"
   THEN VO%=-12
110 D%=0
   :L%=3
   :S%=0
   :N%=0
   :O%=1
120 VDU 23,224,36,126
   ,153,153,255,126,66
   ,231,23,225,60,110
   ,255,85,1,85,255,60
   :VDU 23,226,60,126
   ,255,255,255,255,126
   ,60,23,233,0,0,126
   ,0,0,126,0,0,23,255
   ,28,28,8,127,8,20
   ,34,65
130 VDU 23,227,60,118
   ,255,170,128,170,255
   ,60,23,228,0,13,63
   ,235,235,63,13,0,23
   ,229,0,176,252,215
   ,215,252,176,0,23
   ,230,27,255,175,253
   ,223,178,255,233,23
   ,231,171,255,106,253
   ,223,117,255,219,23
   ,232,0,129,219,255
   ,255,219,129,0
140 V%=VO%
   :CLEAR
   :VO%=V%
   :MODE 5
   :VDU 23;8202;0;0;0
   :PROCINIT
150 PROCSCREEN
   :REPEAT
   :PROCSPLATCHECK
160 PROCMOVEYOU
170 PROCSPLATCHECK
180 PROCMOVESPACRAFT(RO%)
   :RO%=RO%+1
   :IF RO%=10 RO%=0
190 IF (S% DIV 1000)>N%

   THEN L%=L%+1
   :N%=N%+1
200 UNTIL AMOHOME%=4
210 D%=D%+1
   :O%=O%+1
   :IF D%<9
   THEN 140
220 PROCWELLDONE
   :GOTO 110
230 DEF PROCINIT
   :DIM A%(9)
   :RESTORE 700
   :FOR A%=0TO 9
   :READ R%
   :A%(A%)=R%
   :NEXT
   :DIM A$(9)
   :A1$=CHR$ (224)
   :A2$=CHR$ (225)
   :A3$=CHR$ (227)
   :A4$=CHR$ (226)
   :A5$=CHR$ (233)
   :B1$=CHR$ (228)
   :B2$=CHR$ (229)
   :B3$=CHR$ (230)
   :B4$=CHR$ (231)
   :B5$=CHR$ (232)
240 A$(0)=A1$+A1$+A1$+A2$+A
   4$+A4$+A4$+" 3 "+A1$+A5
   $+" 4 "+A5$+"1"+A5$+
   "1"
   :A$(1)=" 3 "+A1$+" 6
    "+A1$+"1"+A1$+"  9
     "+A4$+A4$+A4$+A4$+A4$
   +A3$+"1"+A1$
250 A$(2)=A4$+A4$+A4$+A4$+A
   4$+A4$+A4$+A4$+"2"+A5$
   +A5$+A5$+"2"+A2$+A4$+A
   4$+A4$+A4$
   :A$(3)=A4$+A3$+"2"+A1$
   +A5$+" 3 "+A1$+A1$+
   "     "+A4$+A4$+A4$+A4$
260 A$(4)=B4$+B4$+B3$+B4$+B
   3$+B3$+B5$+"  "+B1$+B3$
   +B4$+B3$+B5$+"     "+B1$
   +B3$
   :A$(5)=B4$+B3$+B3$+B4$+
   B4$+B3$+B4$+B4$+B3$+B4$
   +B3$+B3$+B2$+"   "+B5$+
   B2$+" "+B5$
270 A$(6)=B1$+B4$+B3$+B5$+
   "   "+B1$+B5$+"   "+B1$
   +B4$+B3$+B4$+B4$+B3$+B5
   $+" "
   :A$(7)="     "+B5$+B4$+
   B3$+B2$+"     "+B5$+B3$
   +B3$+B4$+B3$+B2$
280 A$(8)="    "+B1$+B3$+B4
   $+B3$+B3$+B5$+"
     "+B1$+B3$+B4$+B4$+B5$
   :A$(9)="    "+B5$+B2$+
   "    "+B5$+B3$+B2$+
   "    "+B5$+B2$+" "
290 XX%=10
   :Y%=27
   :LIFE%=3
   :AMOHOME%=0
   :RO%=0
   :DIM AMO%(3)
   :ENDPROC
300 DEF PROCSCREEN
   :VDU 20
   :PRINT TAB(0,0);
   :COLOUR 135
   :PRINT STRING$(40
   ," ")
   :FOR A%=0TO 15STEP 5
   :PRINT TAB(A%,2)" ";
   TAB(A%,3);" ";TAB(A%+4
   ,2);" ";TAB(A%+4,3);
   " "
   :NEXT
   :PRINT TAB(0,17);
   STRING$(20," ")
   :PRINT TAB(0,27);
   STRING$(40," ");
310 PRINT " ";TAB(0,5);
   :COLOUR 128
   :FOR A%=9TO 4STEP -1
   :COLOUR A%(A%)
   :PRINT A$(A%)
   :NEXT
   :COLOUR 7
   :PRINT
   :FOR A%=3TO 0STEP -1
   :COLOUR A%(A%)
   :PRINT A$(A%)
   :NEXT
320 PRINT TAB(1,29);"SCORE:
   ";S%;" LIVES=";L%
330 A$=STRING$(0%,CHR$ (224
   ))
   :COLOUR RND(2)
   :PRINT TAB(0,17);A$;
   TAB(20-LEN (A$),17);A$
   :ENDPROC
340 **********************
   **********
350 DEF PROCMOVEYOU
   :PROCDELETE
   :IF INKEY (-1)
   THEN Y%=Y%+2
   :S%=S%-10
   :SOUND 1,VO%,25,2
   ELSE IF INKEY (-2)
   THEN Y%=Y%-2
   :S%=S%+10
```

```
    :SOUND 1,VO%,75,2
360 FOR A=0TO (10-D%)*4
    :NEXT
370 IF INKEY (-1) OR
    INKEY (-2)
    THEN 390
380 IF INKEY (-102)
    THEN X%=X%+1
    :SOUND 1,VO%,100,1
    :SOUND 1,VO%,60,1
    ELSE IF INKEY (-86)
    THEN X%=X%-1
    :SOUND 1,VO%,60,1
    :SOUND 1,VO%,100,1
390 IF X%>19
    THEN X%=19
    ELSE IF X%<1
    THEN X%=1
400 IF Y%>27
    THEN Y%=27
    :S%=S%+10
    ELSE IF Y%<4
    THEN PROCHOME
410 GOTO 440
420 IF VO%=-12
    THEN RESTORE 430
    :FOR A%=0TO 11
    :READ P%,DU%
    :SOUND 1,0,0,1
    :SOUND 1,-15,P%,DU%
    :NEXT
    ELSE FOR A=0TO 2000
    :NEXT
430 DATA 110,5,110,5,80
    ,10,100,5,100,5,70
    ,10,55,5,40,5,25,5
    ,10,5,150,10
440
450 IF Y%<4
    THEN Y%=27
    :X%=10
460 COLOUR 11
    :PRINT TAB(X%,Y%);
    CHR$ (255)
    :COLOUR 7
    :PRINT TAB(1,29);"SCORE
    :";S%;" LIVES=";L%
470 ENDPROC
480 DEF PROCMOVESPACRAFT(A%
    )
    :IF A% MOD 2 =0 A$(A%)=
    RIGHT$(A$(A%),19)+
    LEFT$(A$(A%),1)
    ELSE A$(A%)=RIGHT$(A$(A
    %),1)+LEFT$(A$(A%)
    ,19)
490 PROCSPA(A%)
    :PROCMOVEYONSPA
    :ENDPROC
500 DEF PROCSPLATCHECK
    :IF Y%=27
    THEN ENDPROC
510 IF Y%=17 AND X%>D%-1
     AND X%<20-D%
    THEN ENDPROC
520 IF Y%>17
    THEN M%=(27-Y%)/2
    ELSE M%=((27-Y%)-2)/2
    :GOTO 550
530 IF MID$(A$(M%-1),X%+1
    ,1)<>" "
    THEN 570
540 ENDPROC
550 IF MID$(A$(M%-1),X%+1
    ,1)=" " AND MID$(A$(M%-
    1),X%+(-X%)+1,1)=
    " "
    THEN 570
560 ENDPROC
570 PROCDEAD
    :GOTO 160
580 DEF PROCDEAD
    :L%=L%-1
    :IF L%=0
    THEN PROCEND
590 IF Y%<4 OR Y%>18
    THEN 620
    ELSE VDU 23,252,17
    ,18,212,248,212,18
    ,17,0
    :VDU 23,253,65,34
    ,20,8,127,8,28,28
    :VDU 23,254,136,72
    ,43,31,43,72,136,0
    :P%=100
600 REPEAT
    :FOR A%=252TO 255
    :COLOUR RND(3)+1
    :PRINT TAB(X%,Y%);
    CHR$ (A%)
    :FOR B%=0TO P%/4
    :VDU 19,(B%+1)MOD 6+1
    ,RND(7),0,0,0
    :FOR C%=0TO 10
    :NEXT
    :VDU 19,(B%+1)MOD 6+1
    ,(B%+1)MOD 6+1,0,0
    ,0
    :NEXT
    :SOUND 1,VO%,P%*2+25
    ,5
    :P%=P%-2.5
    :NEXT
    :UNTIL P%<-20
    :*FX15
610 SOUND 0,VO%,4,30
    :FOR A%=0TO 40
    :FOR B%=1TO 7
    :VDU 19,B%,8-B%,0
    ,0,0
    :NEXT
    :FOR B%=0TO 20
    :NEXT
    :FOR C%=1TO 7
    :VDU 19,C%,C%,0,0
    ,0
    :NEXT
    :NEXT
    :X%=10
    :Y%=27
    :PROCSCREEN
    :ENDPROC
620 VDU 23,254,0,28,28
    ,8,127,8,20,34,23
    ,253,0,0,28,28,8,127
    ,8,20,23,252,0,0,0
    ,28,28,8,127,8,23
    ,251,0,0,0,0,28,28
    ,8,127,23,250,0,0
    ,0,0,0,28,28,8,23
    ,249,0,0,0,0,0,0,28
    ,28,23,248,0,0,0,0
    ,0,0,0,28
    :IF Y%<4
    THEN Y%=Y%+1:
630 FOR A%=255 TO 248
    STEP -1
    :PRINT TAB(X%,Y%);
    CHR$ (A%)
    :COLOUR RND(7+6)
    :FOR A=0TO 100
    :NEXT ,
    :PRINT TAB(X%,Y%);
    " ";
    :X%=10
    :Y%=27
    :SOUND 0,VO%,4,20
    :FOR A%=0TO 10
    :FOR B%=1TO 7
    :VDU 19,B%,RND(7)
    ,0,0,0
    :NEXT
    :FOR B%=0TO 10
    :NEXT
    :FOR B%=1TO 7
    :VDU 19,B%,B%,0,0
    ,0
    :NEXT ,
640 PROCSCREEN
    :ENDPROC
650 DEF PROCEND
    :CLS
    :A$=STRING$(80,
    CHR$ (255))
    :PRINT '''A$
    :B$=CHR$ (225)+
    CHR$ (226)+CHR$ (226)+
    CHR$ (226)+CHR$ (226)+
    CHR$ (226)+CHR$ (226)+
    CHR$ (226)+CHR$ (226)+
    " "
    :FOR A%=160TO 80
    STEP -2
    :PRINT TAB(A%MOD 20
    ,A%DIV 20);B$
    :SOUND 1,VO%,100,2
660 FOR A=0TO 250
    :NEXT
    :NEXT
    :FOR A%=0TO 12
    :A$=STRING$(A%," ")+
    "BAD LUCK"
    :PRINT A$
    :B%=RND(100)+100
    :SOUND 1,VO%,B%,2
    :SOUND 2,VO%,B%+1
    ,2
    :SOUND 3,VO%,B%+2
    ,2
    :FOR B%=0TO 100
    :NEXT
    :NEXT
    :PRINT "You are dead!!"
    '''"You were on level
     ";D%;'"when you got
     killed"
670 IF S%>H%
    THEN H%=S%
680 PRINT '''"Your SCORE
     was ";S%
    :PRINT "HISCORE is
     ";H%
    :*FX15
690 PRINT "Another Game
     Y/N"
    :G$=GET$
    :IF G$="Y"
    THEN 40
    ELSE IF G$="N"
    THEN END
    ELSE SOUND 1,VO%,64
    ,5
    :GOTO 690
700 DATA 3,1,2,3,1,2,3
    ,1,2,3
710 DEF PROCWELLDONE
    :CLS
    :VDU 23,252,7,7,7
    ,1,63,63,55,55,23
    ,253,224,224,224,128
    ,252,252,236,236,23
    ,254,7,7,7,7,6,6,14
    ,14,23,255,224,224
    ,224,224,96,96,112
    ,112
```

Space Hike listing

From Page 55

```
720 PRINT TAB(9,5);
    CHR$ (252);CHR$ (253);
    TAB(9,6);CHR$ (254);
    CHR$ (255);TAB(8,3);
    "YOU!"
    :FOR A=0TO 1000
    :NEXT
    :PRINT TAB(8,3);"     "
730 A$=CHR$ (225)+CHR$ (226
    )+CHR$ (226)+CHR$ (226)
    +CHR$ (226)+CHR$ (226)+
    CHR$ (226)+CHR$ (226)+
    CHR$ (226)+" "
    :B$=" "+CHR$ (226)+
    CHR$ (226)+CHR$ (226)+
    CHR$ (226)+CHR$ (226)+
    CHR$ (226)+CHR$ (226)+
    CHR$ (226)+CHR$ (227)
    :FOR A%=160TO 131
    STEP -1
740 PRINT TAB(A%MOD 20
    ,A%DIV 20);A$
    :SOUND 1,-13,101,2
    :FOR A=0TO 250
    :NEXT
    :NEXT
    :FOR A%=0TO 300
    :PRINT TAB(11,5);"!";
    TAB(11,5);" "
    :NEXT
    :FOR A%=0TO 30
    :PRINT TAB(11,6);
    CHR$ (227);
    :FOR A=0TO 100
    :NEXT
    :VDU 8,225
    :FOR A=0TO 100
    :NEXT
    :NEXT
    :FOR A%=131TO 300:
750 PRINT TAB(A%MOD 20
    ,A%DIV 20);B$
    :SOUND 1,-13,255,1
    :FOR A=0TO 100
    :NEXT
    :NEXT
760 PRINT '"" CONGRATULATI
    ONS!""   YOU HAVE
     BEATEN""     THE
     BBC MICRO!!"
    :FOR A%=0TO 10000
    :NEXT
    :ENDPROC
770 DEF PROCHOME
780 IF X%>0 AND X%<4
    AND AMO%(0)=0 PRINT
    TAB(1,2);
    :VDU 255,255,255,8
    ,8,8,10,255,255,255
    :AMOHOME%=AMOHOME%+1
    :AMO%(0)=-1
    :GOTO 420
790 IF X%>5 AND X%<9
    AND AMO%(1)=0 PRINT
    TAB(6,2);
    :VDU 255,255,255,8
    ,8,8,10,255,255,255
    :AMOHOME%=AMOHOME%+1
    :AMO%(1)=-1
    :GOTO 420
800 IF X%>10 AND X%<14
     AND AMO%(2)=0
    PRINT TAB(11,2);
    :VDU 255,255,255,8
    ,8,8,10,255,255,255
    :AMOHOME%=AMOHOME%+1
    :AMO%(2)=-1
    :GOTO 420
810 IF X%>15 AND X%<19
     AND AMO%(3)=0
    PRINT TAB(16,2);
    :VDU 255,255,255,8
    ,8,8,10,255,255,255
    :AMOHOME%=AMOHOME%+1
    :AMO%(3)=-1
    :GOTO 420
820 PROCDEAD
    :ENDPROC
830 DEF PROCDELETE
    :A%=7
840 IF Y%=27 OR Y%=17
    COLOUR 135
    ELSE COLOUR 128
850 IF Y%>15
    THEN S$=" "
    ELSE A%=10-((Y%-1)/2-1)
    :S$=MID$(A$(A%),X%+1
    ,1)
860 COLOUR A%(A%)
    :PRINT TAB(X%,Y%);S$
    :COLOUR 128
    :COLOUR 7
    :ENDPROC
870 DEF PROCSPA(A%)
    :IF A%<4
    THEN Z%=6-(A%*2)+19
    ELSE Z%=20-(A%*2)+3
880 COLOUR A%(A%)
    :PRINT TAB(0,Z%);A$(A%)
    :ENDPROC
890 DEF PROCMOVEYONSPA
900 IF A% MOD 2=0
    THEN X5%=-1
    ELSE X5%=1
910 IF Y%=Z% AND Y%<17
    THEN X%=X%+X5%
920 ENDPROC
```

This listing is included in this month's cassette tape offer. See order form on Page 43

Get your message taped

DID you know that there's a simple way to use your Electron as a kind of typewriter? Instead of writing on paper you write your message into the Electron's memory and save it onto a cassette tape.

You can then send the tape to someone with an Electron or a BBC Micro and they can load the message into their micro and read it.

It's remarkably simple. All you do is to type in the message you want from the keyboard just as though you were typing in a program.

You enter the line number as normal and then start typing the message. When you've written enough on one line, press Return, then enter a new line number and start typing again.

It's simple and it's easy, and it allows you to use the micro as a very, very elementary word processor.

Of course, all the usual program editing facilities still apply. If you don't like line 30 you can get rid of it all by typing in 30 and pressing Return to get rid of the lot.

Or you could use the cursor and Copy keys to alter the old version. You can edit it just as though it were a normal program.

But you can't RUN it – you get an error message. That, though, is no problem, as we don't want to run what we've written. We just want to save it on tape so we can send it to someone.

This we do in the normal way, just using a file name like:

SAVE "MESSAGE"

To the micro it's just another program so it saves it to tape like any other program. You can then send it to whoever you want. They LOAD it just like a normal program and read the message, ignoring the line numbers.

As I said before, it's very simple and very useful as people who have word processors on their BBC Micros will be able to load your program, get rid of the line numbers and print it out on a printer.

Of course, these same facilities will come to the Electron eventually.

So, if you want to send messages, send them as a program! And if you want to send an article to *Electron User* but don't have a typewriter, then send us the message disguised as a program.

We'll do the rest.

```
 10 This is an example of how to
 20
 30 write using your Electron as
 40
 50 a typewriter. I've only left
 60
 70 the spaces in between the lines
 80
 90 for clarity. Also the lines can
100
110 be a lot longer, up to 255
120
130 characters. Using this method
140
150 you can send legible messages
160
170 on cassette tape.
```

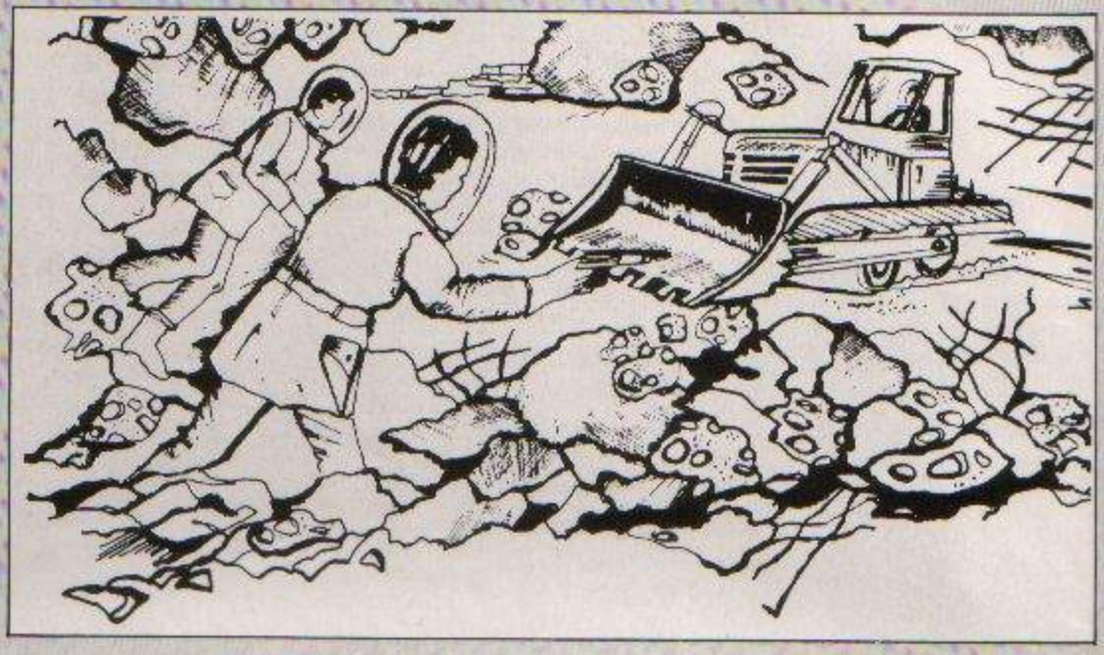
From Page 35

This listing was produced using a special formatter which breaks one program line over several lines of listing. When entering a line don't press Return until you come to the next line number. Full details of the formatter are given on Page 4 of the February issue.

```
170 VDU 4
    :PRINT TAB(0,0)"    "
    :VDU 5
180 ENDPROC
190 REM **** Crash ****
200 DEF PROCCrash
210 VDU 5,19,3,11;0;
220 FOR J%=100TO 500STEP 2
230 GCOL 3,RND(3)
    :MOVE RND(J%)-J%DIV 2
    ,RND(J%)-J%DIV 2
    :VDU 249
    :SOUND 16,-15,RND(3)+3
    ,40
240 NEXT
250 TIME =0
    :REPEAT UNTIL TIME >200
260 G%=G%-1
    :F%=-1
270 ENDPROC
280 REM **** Hit ****
290 DEF PROCHit
300 IF ABS (X%(I%)+16)>40
    OR ABS (Y%(I%)-64)>32
    THEN PROCCrash
    :ENDPROC
310 X%(I%)=-16
    :Y%(I%)=48
    :PROCPlot
    :Y%(I%)=2000
    :D%=D%+5
320 IF T%(I%)=1
    THEN A%=A%+10
330 IF T%(I%)=2
    THEN A%=A%+50
340 IF T%(I%)=6
    THEN PROCBaddy
350 VDU 4
    :PRINT TAB(11,0);A%
    :VDU 5
360 ENDPROC
370 REM **** Initial ****
380 DEF PROCInitial
390 DIM X%(5),Y%(5),DX%(5)
    ,DY%(5),T%(5)
400 VDU 23,240,64,32,128
    ,208,160,208,64,96,23
    ,249,32,112,168,80,80
    ,168,0,80,23,250,65
    ,65,34,62,127,127,73
    ,28
410 VDU 23,241,0,0,64,96
    ,96,32,48,48,23,242
    ,0,0,2,6,6,4,12,12
420 VDU 23,243,16,16,24
    ,24,152,156,156,158
    ,23,244,8,8,24,24,25
    ,57,57,121
430 VDU 23,245,191,191,255
    ,255,255,255,255,255
    ,23,246,253,253,255
    ,255,255,255,255,255
440 VDU 23,247,195,195,135
    ,135,7,15,14,14,23,248
    ,195,195,225,225,224
    ,240,112,112
450 ENVELOPE 1,2,1,1,-1
    ,10,20,18,126,0,0,-126
    ,126,126
460 ENDPROC
470 REM **** Instruct ****
480 DEF PROCInstruct
490 PRINT TAB(10,0)"ASTEROID
    PROSPECTOR"TAB(10,1)
    "~~~~~~~~~~~~~~~~~~~~"
    :*FX9,5
500 PRINT TAB(3,2)"Your task
    is to collect a valuable
    mineral found in aster
    oids. Red ones contain
    small quantities of
    the mineral (10 points)
    but green ones contain
    much more. (50 points)"
    :*FX10,5
510 PRINT "  Your task is
    dangerous! Your ship
    canbe mortally damaged
    if asteroids collidewith
    the ship's side. The
    really bad ones are
    the mines! These have
    been set to protect
    the asteroid swarm from
    rival";
520 PRINT "prospectors. Shoul
    d you collect one of
    these, you will have
    only a few seconds to
    defuse it by typing
    in the three letter
    combination displayed
    at the top left of the
    screen. If you mistype
    it"
530 PRINT "or are too slow
    then you blow up!"
    TAB(15,19)"CONTROLS"
    TAB(15,20)"~~~~~~~~"
    TAB(6,21)"> .....Rotate
    ship to right
          < .....Rotate
    ship to left
          Z .....Thrust"
540 PRINT TAB(10,24)"PRESS
    SPACE TO START";
    :REPEAT UNTIL GET =32
550 ENDPROC
560 REM **** Moveast ****
570 DEF PROCMoveast
580 C%=T%(I%)AND 3
    :PROCPlot
590 X%(I%)=X%(I%)+DX%(I%)
    :Y%(I%)=Y%(I%)+DY%(I%)
    :IF R%=0
    THEN K=1
    :L=0
    ELSE K=0.997858923
    :L=6.54031292E-2
    :IF R%=-1
    THEN L=-L
600 PROCRotate
    :IF S%=1
    THEN Y%(I%)=Y%(I%)-48
610 IF ABS (X%(I%)+16)<80
    AND ABS (Y%(I%)-12)<78
    THEN PROCHit
    ELSE PROCPlot
620 IF ABS (X%(I%))>1000
    OR ABS (Y%(I%))>1000
    THEN X%(I%)=SGN (X%(I%))*
    700
    :Y%(I%)=-SGN (Y%(I%))*700
    :DX%(I%)=RND(65)-33
    :DY%(I%)=RND(65)-33
    :T%(I%)=RND(3)
    :IF T%(I%)=3
    THEN T%(I%)=6
630 ENDPROC
640 REM **** Play ****
650 DEF PROCPlay
660 F%=0
    :REPEAT
    :R%=INKEY (-104)-
    INKEY (-103)
    :S%=-INKEY (-98)
    :VDU 5
    :MOVE -16,48
    :GCOL 0,0
    :VDU 240
    :PROCRocket
    :IF S%=1OR R%<>0
    THEN SOUND 16,-15,6
    ,20
670 FOR I%=1TO 5
680 PROCMoveast
690 NEXT
700 PROCRocket
710 UNTIL F%
    :IF G%<>0
    THEN F%=0
720 ENDPROC
730 REM **** Plot ****
740 DEF PROCPlot
750 IF ABS (X%(I%)+16)>624
    OR ABS (Y%(I%)+16)>482
    THEN ENDPROC
760 MOVE X%(I%),Y%(I%)
    :GCOL 3,C%
    :VDU 240
770 ENDPROC
780 REM **** Rocket ****
790 DEF PROCRocket
800 GCOL 3,1
    :IF R%=1
    THEN MOVE 40,-48
    :VDU 249
810 IF R%=-1
    THEN MOVE -80,-48
    :VDU 249
820 IF S%=1
    THEN MOVE -40,-64
    :VDU 249
    :MOVE 0,-64
    :VDU 249
830 ENDPROC
840 REM **** Rotate ****
850 DEF PROCRotate
860 X%(I%)=INT (X%(I%)*K+Y%(I
    %)*L+.5)
    :Y%(I%)=(Y%(I%)*K-X%(I%)*
    L+.5)
870 DX%(I%)=INT (DX%(I%)*K+DY
    %(I%)*L+.5)
    :DY%(I%)=INT (DY%(I%)*K-D
    X%(I%)*L+.5)
880 ENDPROC
```

Asteroids listing

From Page 57



```
890 REM **** Setup ****
900 DEF PROCSetup
910 VDU 5,29,640;498;
920 GCOL 0,3
    :MOVE -64,64
    :VDU 5,241,242,8,8,10
    ,243,244,8,8,10,245
    ,246,8,8,10,247,248
    ,4
930 FOR I%=1TO 5
    :X%(I%)=700*SGN (
    RND(2)-1.5)
    :Y%(I%)=700*SGN (
    RND(2)-1.5)
    :DX%(I%)=RND(D%)-D%
    DIV 2
    :DY%(I%)=RND(D%)-D%
    DIV 2
    :T%(I%)=RND(2)
    :NEXT
940 VDU 19,2,2;0;19,3,6;0;
950 COLOUR 2
    :PRINT TAB(5,0)"SCORE
    ";TAB(16,0);
    :COLOUR 3
    :FOR I=1TO 6%
    :VDU 250
    :NEXT
    :COLOUR 0
    :VDU 8,250,250
    :COLOUR 1
    :PRINT TAB(11,0);A%
960 ENDPROC
```

This listing is included in this month's cassette tape offer. See order form on Page 43.

Don't be fooled – this fast and furious game by MARK SMIDDY is trickier than it looks . . .

It's the BUNNY BLITZ!

EASTER is almost here and with it the *Electron User* Bunny Blitz. It's a simple little game to learn, but one that's fiendish to play.

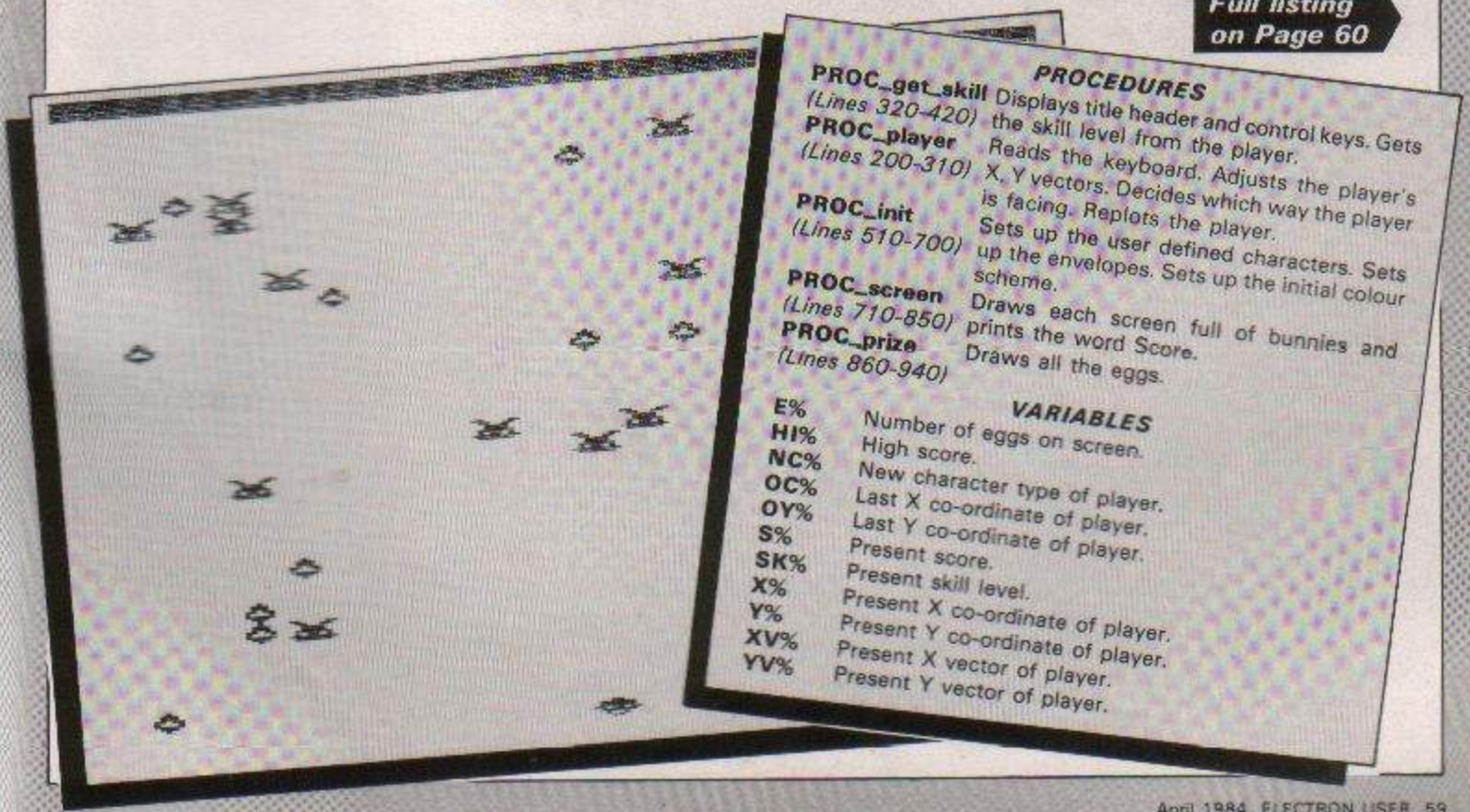
You have to dash around collecting Easter eggs that are scattered about the screen. The trouble is that once you start going you can't stop.

You gain points for every egg you collect but whenever you bump into a bunny points are lopped off your total.

Not only that, but you are fighting against the clock.

Clear a screen in time and you are faced with another screen containing even more bunnies. Such is life in the world of the microchip.

Full listing on Page 60



PROCEDURES

PROC_get_skill *(Lines 320-420)* Displays title header and control keys. Gets the skill level from the player.

PROC_player *(Lines 200-310)* Reads the keyboard. Adjusts the player's X, Y vectors. Decides which way the player is facing. Replots the player.

PROC_init *(Lines 510-700)* Sets up the user defined characters. Sets up the envelopes. Sets up the initial colour scheme.

PROC_screen *(Lines 710-850)* Draws each screen full of bunnies and prints the word Score.

PROC_prize *(Lines 860-940)* Draws all the eggs.

VARIABLES

E% Number of eggs on screen.
HI% High score.
NC% New character type of player.
OC% Last X co-ordinate of player.
OY% Last Y co-ordinate of player.
S% Present score.
SK% Present skill level.
X% Present X co-ordinate of player.
Y% Present Y co-ordinate of player.
XV% Present X vector of player.
YV% Present Y vector of player.

Bunny Blitz listing

This listing was produced using a special formatter which breaks one program line over several lines of listing. When entering a line don't press Return until you come to the next line number. Full details of the formatter are given on Page 4 of the February issue.

From Page 59

```
 1 REM BUNNY BLITZ
 2 REM MARK SMIDDY
 3 REM (C) ELECTRON USER

 10 REPEAT
 20 MODE 1
 30 PROC_getskill
 40 MODE 2
 50 VDU 23,0,8202;0;0;0;
 60 PROC_init
 70 TIME =0
 80 REPEAT
 90 REPEAT
100 VDU 4
    :PRINT TAB(6,0);S%
    :VDU 5
110 X%=X%+XV%
    :Y%=Y%+YV%
120 PROC_player
130 IF POINT(X%+16,Y%-12)=5
    SOUND &11,1,100,3
    :MOVE X%,Y%-4
    :GCOL 3,1
    :VDU 5,229
    :S%=S%+40
    :E%=E%-1
140 IF POINT(X%+16,Y%-12)=6
    SOUND &10,1,2,2
    :S%=S%-5
150 UNTIL TIME >=6000
    OR E%=0
151 IF E%=0 CLS
    :PROC_screen
    :TIME =0
    :IF SK%>7 SK%=SK%-5
160 UNTIL TIME >=6000
170 MODE 1
180 PROC_result
190 UNTIL 0
200 DEF PROC_player
210 VDU 5
    :GCOL 3,4
220 IF INKEY (-66) YV%=32
    :XV%=0
    :NC%=3
230 IF INKEY (-98) YV%=-32
    :XV%=0
    :NC%=2
240 IF INKEY (-103) XV%=-64
    :YV%=0
    :NC%=0
250 IF INKEY (-104) XV%=64
    :YV%=0
    :NC%=1
260 IF X%>=1216 X%=1216

    ELSE IF X%<=0 X%=0
270 IF Y%>=992 Y%=992
    ELSE IF Y%<32 Y%=32
280 MOVE OX%,OY%
    :VDU 224+OC%
290 MOVE X%,Y%
    :VDU 224+NC%
300 OX%=X%
    :OY%=Y%
    :OC%=NC%
310 ENDPROC
320 DEF PROC_getskill
330 PRINT TAB(9,1)"Welcome
    to Bunny Blitz"
340 PRINT TAB(14,4)"A
    = Up";TAB(14,6)"Z
    = Down"
350 PRINT TAB(14,8)"<
    = left";TAB(14,10)
    ">  =  Right"
360 REPEAT
365 PRINT TAB(0,20) "2
    is the hardest level,
    40's the easiest"
370 INPUT TAB(0,16)"Choose
    your concentration
    DOC' (2/40) "SK%
380 UNTIL SK%>=2 AND SK%<=40
390 PRINT TAB(0,16)
    STRING$(40," ")
    TAB(0,16)"Press the
    SPACE BAR to play"
400 *FX15,1
410 REPEAT UNTIL 32=GET
420 ENDPROC
430 DEF PROC_result
440 VDU 4
    :COLOUR 2
    :COLOUR 128
450 IF HI%>=S% HI%=S%
455 PRINT TAB(10,1)"SORRY
    out of time!!!"
460 PRINT TAB(10,3)"Bunny
    Blitz Results"
470 PRINT '''"Final score
    ";S%''"High Score
    ";HI%
480 *FX15,1
490 PRINT TAB(0,30);"Press
    SPACE for a new game"
    :REPEAT UNTIL 32=GET
500 ENDPROC
510 DEF PROC_init
520 XV%=0
    :YV%=0
530 X%=640
    :Y%=544
540 OX%=640
    :OY%=544
550 NC%=0
    :OC%=0
560 S%=0
    :HI%=0
    :E%=0
570 VDU 23,224,0,24,126
    ,183,126,110,60,24
580 VDU 23,225,0,24,126
    ,237,126,118,60,24
590 VDU 23,226,0,24,126
    ,126,219,126,36,24
600 VDU 23,227,0,24,90
    ,255,126,102,60,24
610 VDU 23,228,195,102
    ,60,60,126,90,255
    ,126
620 VDU 23,229,16,56,56
    ,108,68,124,56,16
630 VDU 19,0,4;0;
640 VDU 19,1,3;0;
650 VDU 19,2,2;0;
660 VDU 19,3,0;0;
661 VDU 19,4,7;0;
670 ENVELOPE 1,1,8,-8
    ,8,4,4,4,126,0,0,-126
    ,126,126
680 ENVELOPE 2,1,20,-20
    ,20,45,45,45,126,0
    ,0,-126,126,126
690 PROC_screen
700 ENDPROC
710 DEF PROC_screen
720 VDU 4
    :COLOUR 2
    :COLOUR 131
730 PRINT TAB(0,0)STRING$(2
    0," ")
740 PRINT TAB(0,0)"Score:"
750 COLOUR 2
    :COLOUR 128
760 LOCAL X%,Y%
770 FOR X%=0TO 18
780 FOR Y%=1 TO 30
790 PRINT TAB(X%,Y%);
800 IF RND(SK%)=1 VDU 228
810 NEXT
    :NEXT
820 PROC_prize
830 COLOUR 131
840 VDU 5
    :MOVE OX%,OY%
    :
    :GCOL 3,4
    :VDU 224
    :VDU 4
850 ENDPROC
860 DEF PROC_prize
870 COLOUR 1
880 LOCAL X%,Y%
890 FOR X%=0TO 18
900 FOR Y%=1 TO 30
910 PRINT TAB(X%,Y%);
920 IF RND(SK%)=1 VDU 229
    :E%=E%+1
930 NEXT
    :NEXT
940 ENDPROC
```

This listing is included in this month's cassette tape offer. See order form on Page 43.

Micro Messages

*FX 210,1 for the sound of silence...

SOME of the programs I have for my Electron are on the noisy side, to say the least! But I find that sometimes I prefer the sound to be quieter or not on at all.

Is there any way I can do this without adjusting the program listing? – **Gary Cunningham, Stockport.**

● You can switch off the Electron's sound channels by typing in *FX 210,1 before you run the program. To get the sound back again you use *FX 210,0.

Spacings problem

MY family enjoy typing in the programs in "Electron User" but you will have to do something about your listings!

The biggest problem is with program lines which contain spaces which are broken over more than one line.

It is impossible to know how many spaces to key in.

Wouldn't it be possible for you to add a character into your listing, like an underline instead of a space, so that we could count them?

The listing for "Parky" in the March issue was a nightmare to sort out. – **A.L. Beaumont, Clapham.**

● Point taken, Mr Beaumont. We'll try to be more careful in future.

A satisfied customer

I THOUGHT I would take this opportunity to congratulate you on the very clear format of your listings. This is unlike the very obscure printing in many other magazines.

Carry on the good work. – **E.W.W. Theobald, Kings Langley, Herts.**

● Thanks for your letter. It's nice to hear from a satisfied customer. We try to make the listings as clear as possible, though when you make a mistake typing in a program (as we all do) it's much easier to blame the listing than ourselves!

Telling O from 0

HAVING just bought an Electron we have found your magazine a real help — "Listings Loopholes" (February issue) was especially good.

We have also enjoyed your program listings – but just one plea. Please differentiate between O and 0! – **D. Blackburn, Sheffield.**

● We know trying to tell the difference between O and 0 can be quite frustrating at first. Sadly we are limited to the characters on our printer, which does not differentiate them.

However, in a short while it will become second nature to you and your problems will ease. After all O and 0 are reasonably different as the following shows:

OOOOOOOOOOOOOOOOOO
0000000000000000000

Review reviewed

THANK you for reviewing "Start Programming with the Electron" in the December issue of Electron User.

Your concern about the early introduction of procedures and recursion is understandable but we have not just done that by accident.

This was as the result of research by an ex-masters student of Exeter University, with a variety of novice programmers, which showed us the reason pupils have major problems with procedures and recursion is because they are usually introduced as an afterthought in most computing courses.

For example, we discovered that pupils who were introduced to recursion before they saw iteration had less problems conceptualising what recursion was than the ones who learned it after they had been introduced to iteration constructs such as FOR loops.

The mention that we go on to introduce functions in chapter two betrays the fact that your reviewer has not in fact read the book.

Functions are introduced in chapter four, a good distance away from procedures, as we also discovered that when procedures and functions are introduced together, pupils develop a habit of mixing them up with each other.

The two constructs have similar syntax but radically different semantics.

Your mention that the explanation of expression has been inadequate was of great help to us as it confirmed the same evidence we had received from our field trials.

We are currently revising the Start Programming with the Electron book in order to turn it into a book for the BBC Micro and would be grateful for further comments and suggestions for improvements from you and other people who have read the book. – **Masoud Yazdani, University of Exeter.**

● I can assure Mr Yazdani that I have in fact read the book, though I think he knows that already, or why should he say that I'd been "of a great help" with my comments!

If he reads the sentence about procedures and expressions again, he'll see that the point I was trying to make was that procedures, functions and conditional branching (which is in chapter two) come before INPUT command is treated.

He does not answer this point in the letter nor any of the other points raised in the review.

Having said that I must say that I find his research findings most interesting and applaud his efforts to help novice programmers.

Peter Green

Positron poser

HELP! With reference to the Positron Invaders listing (Electron User, February) my Electron tells me there is no such FN/PROC at line 470.

Can you help please? – **Ann Cross, Oxford.**

● Thanks for telling us the error message. You'd be surprised how many times people write in with problems and don't tell us what the Electron is doing!

The Electron is telling you that you've tried to use a function but it

→

DO you like us or do you hate us? Are our games too hard or too easy? And what about the articles?

Write to us at Micro Messages and tell us. We can take it!

Remember, that these are the pages that you write yourselves. So tear yourself away from your Electron keyboard and drop us a line.

The address is:
Micro Messages
Electron User
Europa House
68 Chester Road
Hazel Grove
Stockport
SK7 5NY.

Micro Messages

From Page 61

can't find that function.

We suspect your trouble stems from line 1370 where the function being called in line 470 is defined.

If line 470 has been typed in correctly, then it tells the Electron to use the function found in line 1370.

If you've made an error in this line, it will only be noticed when the Electron processes line 470, hence the slightly misleading error message.

So check line 1370 and you should soon be less frustrated.

Positron alternative

AT my son's request, I changed the control keys for the game Positron Invader and thought your readers may like to have the alternative keys.

All you do is change lines 820, 830 and 860 as follows:

```
820 IF A=&2C AND XL>1
THEN XL=XL-1
830 IF A=&2E AND XL<16
THEN XL=XL+1
860 IF A=&20
THEN PROCFIRE
```

This results in the < key moving the base left, the > key moving it right and the space bar firing the laser.

Also at my son's request, I modified line 280 to read:

```
280 PRINTTAB(0,14);
"X----------------X"
```

which has the effect of drawing a clear landing line on the screen.

Thanks for a thoroughly good magazine. Keep up the good work. – **A.M. Dove, Dumbartonshire.**

Problems of saving

AS an Electron owner, I was wondering whether you can give me any advice on how to save programs.

I have had some difficulty in loading from tapes, but with varying the volume on the tape recorder loading has become easier. I am, however, still getting trouble with saving.

It becomes very frustrating having typed in the program then being unsuccessful in saving it.

Can you give me some ideas as to how I can overcome this difficulty. – **G. Dean, Appledore, North Devon.**

● It's always easier to load than to save. We always test that our micro will actually save a short program successfully before we type in long ones.

This saves a lot of anguish.

It could be that you still haven't got the volume and tone controls correct or possibly the recording heads need cleaning.

Sadly, it could just be that your cassette recorder is incompatible with the Electron.

We hope to carry an article on saving and loading programs in a future issue of *Electron User*.

The right direction

I HAVE owned a BBC Micro for two months now. I have bought The Micro User every month for ages, even before I purchased my micro – It's a fine magazine but tends to be just a little advanced for the newcomer.

I always enjoyed Electron User when it was inside The Micro User as it was written in a way which made it easy to understand.

I just bought the first full issue of Electron User and it is very good.

I find most magazines on computers are written for almost expert computer people. They tend to be over technical.

Yours, which is I feel written for the younger micro user, is a step in the right direction.

I think most kids understand micros better and quicker than us old folks. Articles for us should be very simple and straightforward.

Keep the complicated stuff for the kids who after all get taught it in school.

I will probably continue to purchase Electron User as I almost understand it. The Micro User can do without me for a few years till I know what I am doing.

Keep up the good work. How about an adult version of The Micro User (written in the same way as Electron User) for us? – **N. McPherson, Harrow.**

● Many thanks for your letter. It's nice to know that we're so much better than *The Micro User!* I wonder if all our readers agree with the way we present the magazine.

Please talk to us – we're not snooty!

I HAVE only had my BBC B Micro since Christmas and so far I have only used the short programs.

Did anyone realise when the Tapestry program in the February issue of Electron User is run on the BBC Micro, if any letter key is pressed for a few seconds, instead of the space bar, the pattern will change automatically 12 times and then stop.

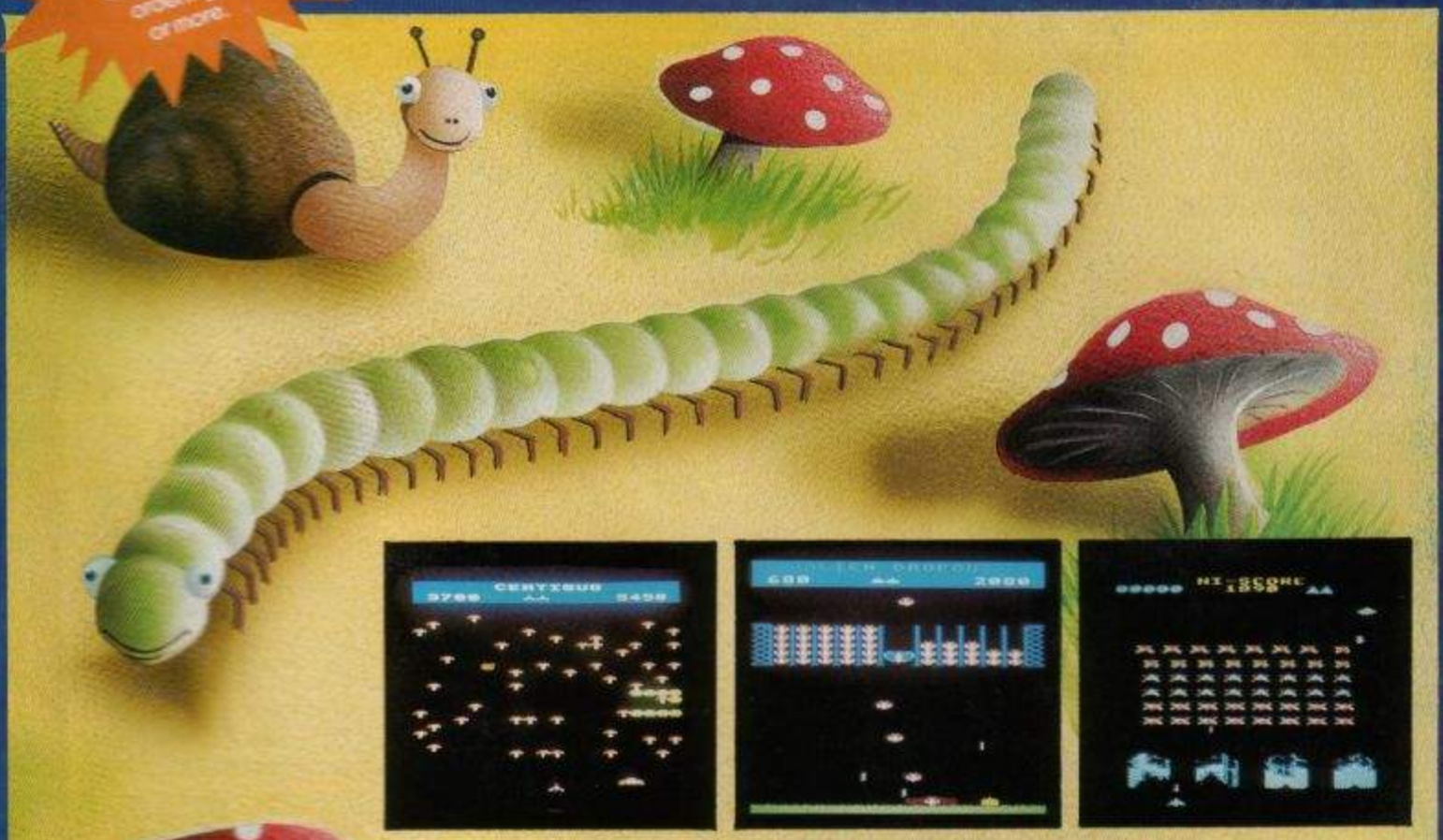
I think your magazine is great for a beginner like me so please tell Pete Bibby that all BBC owners are not "snooty" and that I hope he will still talk to us. – **Michael Smith, Aldershot.**

● Sorry Michael, Pete wasn't really serious when he wrote that. And he says he will talk to BBC Micro owners – he has to because he shares an office with two of them!

SPECIAL OFFER! Deduct £1 per cassette when ordering 2 or more.

TOP QUALITY SOFTWARE FOR THE ACORN ELECTRON

ACORN ELECTRON



CENTIBUG £7.95

The centibug descends from the top of the screen weaving intimidatingly between the mushrooms. Your objective is to shoot all the segments of the centibug before it reaches the bottom of the screen.

Features include: spiders, snails, flies, 6 skill levels, hi-score, rankings, and increasing difficulty.

ALIEN DROPOUT £7.95

A novel and unusual program. Arcade-action with this exciting multi-stage shooting game. The objective of the game is to shoot the aliens out of their "boxes" before the "boxes" fill up. Once full, the aliens fly down relentlessly, exploding as they hit the ground. The game features include: 6 skill levels, rankings, hi-score, increasing difficulty.

INVADERS £7.95

48 marching invaders drop bombs that slowly erode your defences, and two types of spaceship (normal and double speed) fly over releasing large bombs that penetrate through your defences. Increasing difficulty, hi-score, rankings, superb graphics and sound.

WORLD GEOGRAPHY £7.95

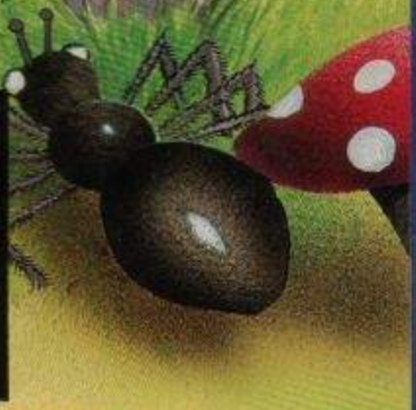
This program covers 166 countries which are divided into 8 categories of difficulty. Each country is pinpointed on an accurate hi-resolution screen map of the world, and the user is asked the capital and/or population. At the end of the test, the percentage of correct answers is given, so that the student can easily monitor his increasing geographical knowledge.

FRUIT MACHINE £7.95

Probably the best fruit machine implementation on the market. This program has it all . . . HOLD, NUDGE, GAMBLE, spinning reels, realistic fruits and sound effects, multiple winning lines. This is THE fruit machine program to buy.

CONSTELLATION £7.95

This fascinating program enables the user to "view the stars" from any point on the Earth's surface, on any date and at any time. A total of 455 stars in 50 constellations may be viewed, and the "telescope" may be moved up, down, left or right, zoomed in or zoomed out. The stars can be displayed by magnitude or constellation.



DISASSEMBLER £7.95

A relocatable disassembler which, unlike some similar programs, allows the disassembled source code to be output to memory. It may then be modified and re-assembled. Other features: page-mode option, output to printer if required, output of RISCII symbols if required.

WE PAY UP TO 20% ROYALTIES FOR HIGH QUALITY BBC MICRO AND ELECTRON PROGRAMS

SUPERIOR SOFTWARE LTD.
Dept. EU4, Regent House,
Skinner Lane, Leeds 7
Tel: 0532 459453

OUR GUARANTEE

1. All our software is available before we advertise.
2. All our software is despatched within 48 hours by first-class post.
3. In the unlikely event that any of our software fails to load, return your cassette to us and we will immediately send a replacement.